CITIZEN COLLEGE

PROJECT NO:

2022-01-53-0153

2022-01-53-0157

A PROJECT

ON

STOCK PRICE PREDICTION USING LSTM

BY

ANURANI KARKI

DENISH ACHARYA

A PROJECT REPORT

SUBMITTED TO THE DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION

IN FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF COMPUTER APPLICATION

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION

LALITPUR, NEPAL

AUGUST, 2024

Stock Price Prediction using LSTM

by

Anurani Karki

Denish Acharya

2022-01-53-0153

2022-01-53-0157

Project Supervisor

Er. Nishan Khanal

A project report submitted in fulfillment of the requirements for the degree of

Bachelor of Computer Application

Department of Bachelor of Computer Application

Pokhara University, Citizen College

Lalitpur, Nepal

AUGUST, 2024

COPYRIGHT ©

DECLARATION

We declare that the work hereby submitted for Bachelor of Computer Application at the Pokhara University, Citizen College entitled "**Stock Price Prediction using LSTM**" is our own work and has not been previously submitted by us at any university for any academic award. We authorize the Pokhara University, Citizen College to lend this project work to other institutions or individuals for the purpose of scholarly research.

Anurani Karki

Denish Acharya

2022-01-53-0153

2022-01-53-0157

AUGUST, 2024

RECOMMENDATION

The undersigned certify that they have read and recommend to the Department of Bachelor of Computer Application for acceptance, a project work entitled "**Stock Price Prediction using LSTM**", submitted by **Anurani Karki and Denish Acharya** in fulfillment of the requirement for the award of the degree of "**Bachelor of Computer Application**".

Project Supervisor
Er. Nishan Khanal
Lecturer/Researcher

BCA Program Coordinator
Er. Nishan Khanal
Department of Bachelor of Computer Application, Citizen College

AUGUST, 2024

DEPARTMENTAL ACCEPTANCE

The project work entitled “**Stock Price Prediction using LSTM**”, submitted by **Anurani Karki and Denish Acharya** in fulfillment of the requirement for the award of the degree of “**Bachelor of Computer Application**” has been accepted as a genuine record of work independently carried out by the student in the department.

Head of the Department
Department of Bachelor of Computer Application,
Citizen College,
Pokhara University, Nepal.

AUGUST, 2024

LETTER OF APPROVAL

We certify that we have examined this report entitled "**Stock Price Prediction using LSTM**", and are satisfied with the **Anurani Karki & Denish Acharya's** Project II. In our opinion, it is satisfactory in the scope and qualifies as a project work in fulfillment of the requirements for the **Bachelor of Computer Application** under **Department of Bachelor of Computer Application, Pokhara University**.

Project Supervisor Er. Nishan Khanal Lecturer/Researcher, Coordinator Citizen College	**Project Board Member** Er. Rajan Bhandari Lecturer/Researcher
Project Board Member Er. Nishanta Sharma Ghimire Lecturer/Researcher	**Project Board Member** Er. Anil Thapa Lecturer/Researcher
Examiner	**Principal** Hari Krishna Aryal Citizen College

AUGUST, 2024

ACKNOWLEDGMENT

This project would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First of all, we would like to express our sincere gratitude to our supervisor, **Er. Nishan Khanal** and co-supervisor, **Bipash Kafle**, of **Citizen College** for providing invaluable guidance, insightful comments, meticulous suggestions, and encouragement throughout the duration of this project work as well as **Pravakar Ghimire** for providing stock data from Nepal stock exchange limited. Our sincere thanks also goes to the BCA coordinator, **Er. Nishan Khanal**, for coordinating the project works, providing astute criticism, and having inexhaustible patience.

We are also grateful to our classmates and friends for offering us advice and moral support. To our family, thank you for encouraging us in all of our pursuits and inspiring us to follow our dreams. We are especially grateful to our parents, who supported us emotionally, believed in us and wanted the best for us.

Anurani Karki
Denish Acharya
2022-01-53-0153
2022-01-53-0157
AUGUST, 2024

ABSTRACT

This project work has endeavored to predict the closing price of the stocks that are traded in the Nepal Stock Exchange using Deep Learning technique. A model was developed that can predict stock prices based on historical closing prices, without any sentiment analysis, with remarkable precision using Long Short-Term Memory networks. LSTM networks, being a type of recurrent neural network, were well-suited for time series prediction due to their ability to capture long-term dependencies in data. The project involved preprocessing stock market data, building and training the LSTM model, and evaluating its performance in different cases. The model was trained on a dataset of a stock closing prices and its predictive accuracy was measured against actual market movements. The model's performance was assessed based on metrics such as Root Mean Square Error, Mean Absolute Error, R Squared, and Mean Absolute Error. It was designed to predict the closing prices for the next 30 days. The results demonstrated that the LSTM model significantly outperformed traditional regression models in predicting stock prices, providing a valuable tool for investors and financial analysts.

Keywords: Long Short-Term Memory, Mean Absolute Error, Prediction, Recurrent Neural Networks, Root Mean Square Error, Stock Price

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADAM	Adaptive Moment Estimation
AI	Artificial Intellegence
ANN	Artificial Neural Network
AR	Auto Regressive
ARCH	AutoRegressive Conditional Heteroskedasticity
ARIMA	Auto Regressive Integrated Moving Average
BPTT	Backpropagation Through Time
CBBL	Chhimek Laghubitta Bittiya Sanstha Limited
CNN	Convolutional Neural Network
CSV	Comma Separated Value
DL	Deep Learning
EMH	Efficient Market Hypothesis
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
IEEE	Institution of Electrical and Electronics Engineers
KPIs	Key Performance Indicators
LSTM	Long Short Term Memory
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MAX	Maximum
MIN	Minimum
ML	Machine Learning
MLP	MultiLayer Perception
MSE	Mean Squared Error
NABIL	Nabil Bank Limited
NEPSE	Nepal Stock Exchange
NLP	Natural Language Processing
NSE	National Stock Exchange
NYSE	New York Stock Exchange

Qty	Quantity
R^2	R squared
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SBL	Siddhartha Bank Limited
SMO	Sequential Minimal Optimization
STD	Standard Deviation
SVM	Support Vector Machine
SVR	Support Vector Regression
UI	User Interface
VS	Visual Studio
WEKA	Waikato Environment for Knowledge Analysis

1 INTRODUCTION

A stock, also known as a share, signifies ownership in a business or corporation held by an individual or group. These shares are listed on stock exchanges, which serve as secondary markets where existing shareholders can trade with potential buyers. The stock market plays a crucial role in the global economy, attracting millions of investors and traders seeking to capitalize on financial opportunities. However, the volatile and unpredictable nature of the stock market poses significant challenges for market participants. Accurate prediction of stock prices is a topic of great interest and importance, as it can potentially lead to better decision-making and improved financial outcomes for investors and businesses. In Nepal, the sole stock exchange is the Nepal Stock Exchange, which began its trading operations on January 13, 1994. NEPSE facilitates transactions through members and intermediaries such as brokers and market makers. Stock market movements are typically driven by investor sentiment, influenced by factors like herd behavior, media stories, rumors, and recommendations. Predicting the market is complex, involving numerous factors including KPIs from quarterly reports, macroeconomic factors, statements from political and influential leaders, news, corporate prospects, global events like currency fluctuations, regulatory changes, market liquidity issues, financial instability, foreign exchange rates, commodity prices, and petroleum product prices. All these elements affect corporate performance and investor sentiment. In Nepal, investment decision-making is generally guided by three major approaches: technical analysis, fundamental analysis, and market sentiment. Technical analysis includes active trading volume, patterns, charts, trends, daily price fluctuations, and historical price data. Fundamental analysis encompasses financial ratios, government regulations, company information, and management quality. Market sentiment involves herd behavior, media stories, rumors, and recommendations. Investors often use a combination of these methods, spending considerable time manually collecting and analyzing data to predict stock movements. However, with the rapid growth of data and computing power, investors are increasingly turning to Artificial Intelligence to identify patterns in real-time transaction and economic data and forecast stock market trends. AI, which involves the simulation of natural intelligence in machines, has gained popularity due to its ability to mimic human actions. Building an AI system involves reverse-engineering human traits and capabilities in a machine, using its computational power to surpass human capabilities.

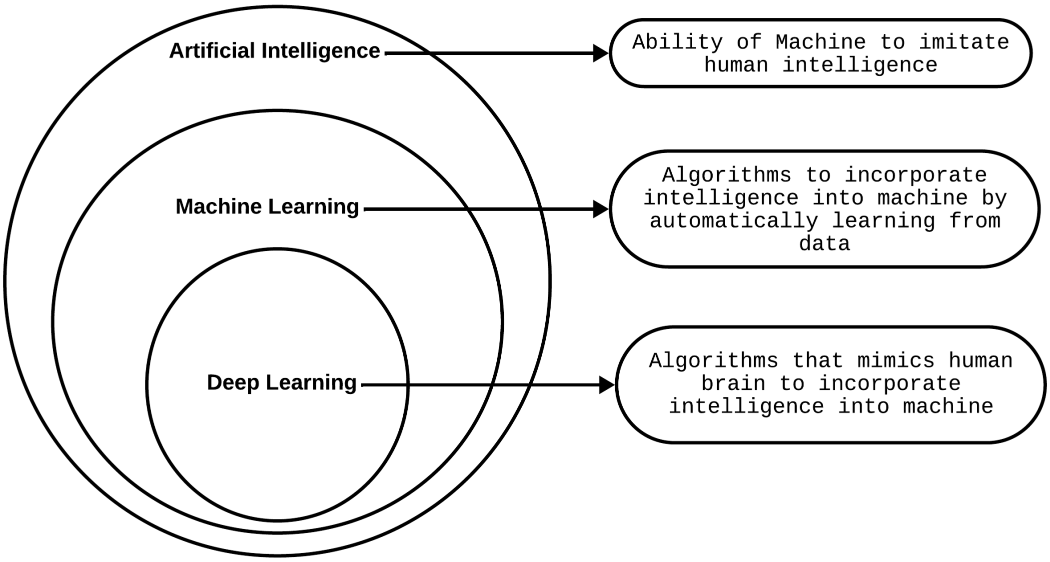

Figure 1.1: Artificial Intelligence vs Machine Learning vs Deep Learning

ML, a subset of AI, uses algorithms to analyze and interpret data, learn from it, and make decisions with minimal human intervention. ML can learn in various ways, including supervised, unsupervised, and semi-supervised learning.

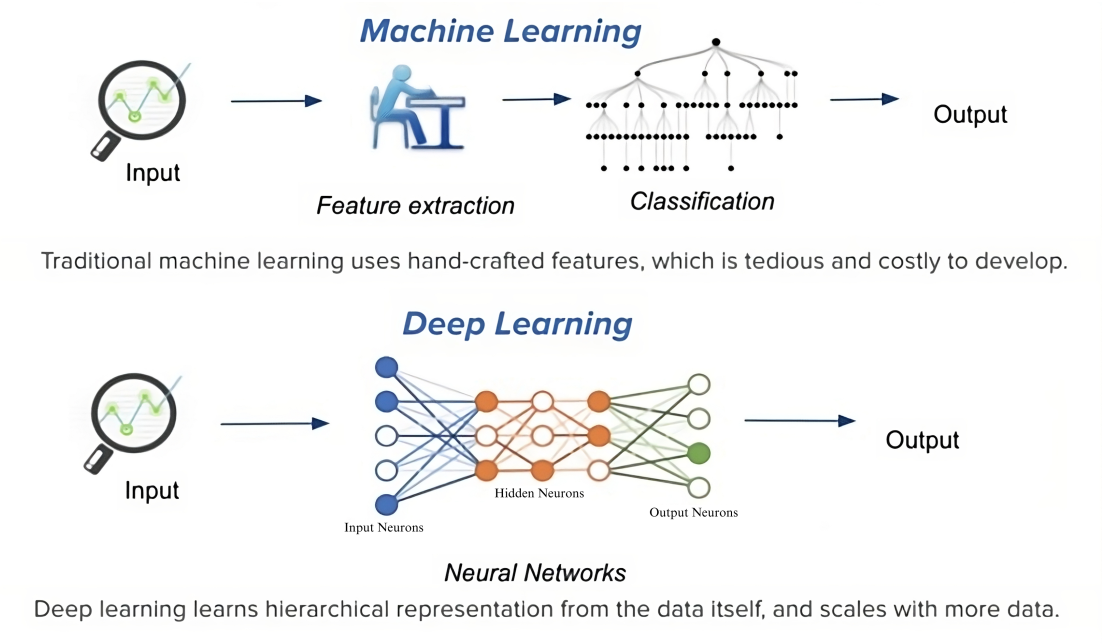

Figure 1.2: Difference between Machine Learning and Deep Learning

DL, a sub-field of machine learning, structures algorithms into multiple layers to create artificial neural networks. These networks can learn from data and make independent decisions. Artificial Neural Networks are complex mathematical models for processing information, inspired by the neurons and synapses in the human brain. Like the brain, a neural network connects simple nodes, or neurons, into a network.

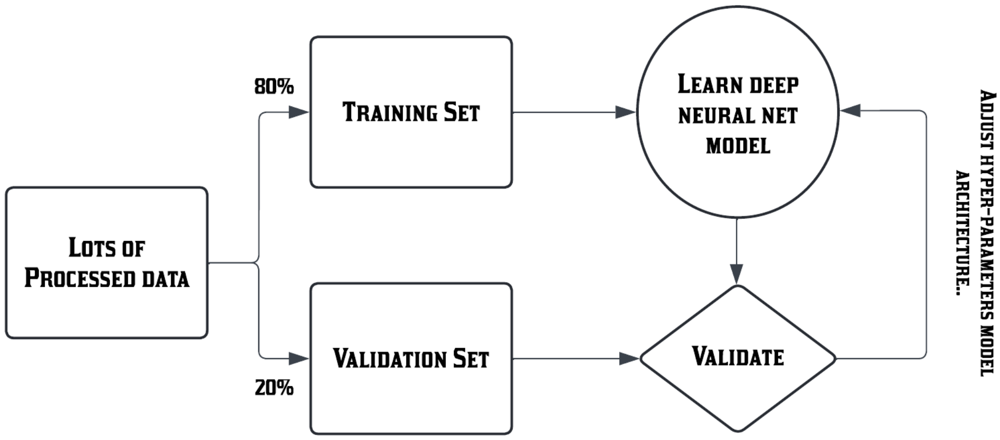

Figure 1.3: Deep Learning Workflow

1.1 Background

The stock market is a complex and dynamic system, where predicting stock prices is a significant challenge due to the myriad factors influencing price movements. Accurate stock price prediction is crucial for investors, businesses, and governments as it aids in informed decision-making, risk assessment, and strategic planning. Traditional methods for predicting stock prices, such as statistical models and technical analysis, often fall short in capturing the intricate patterns and dependencies in financial data. Traditional models, including linear regression and time-series analysis, assume a linear relationship between variables and often fail to account for the non-linear nature of the stock market. These methods are limited in their ability to process large volumes of data and to adapt to rapidly changing market conditions. Consequently, the need for more sophisticated approaches that can handle the complexities and uncertainties inherent in stock market data has led to the exploration of advanced machine learning techniques. In recent years, deep learning has emerged as a powerful tool for analyzing and predicting complex data patterns. LSTM networks, a type of recurrent neural network, have shown particular promise in sequence prediction tasks. LSTMs are designed to capture long-term dependencies in data, making them well-suited for time-series forecasting, including stock price prediction. Unlike traditional methods, LSTMs can learn and model the temporal dependencies in stock prices, leading to more accurate and robust predictions. This system is developed for a robust stock price prediction model using LSTM networks. By leveraging historical market data, the system improve the accuracy of stock price predictions. The methodology involves several key steps, including data preprocessing, feature extraction, model training, and validation. The expected outcome is a predictive model that can generate reliable stock price forecasts, which can be used for various applications such as financial decision-making, algorithmic trading, and market trend analysis.

1.2 Motivation

Figure 1.4: Motivation

The driving force behind this work is the substantial influence of stock price prediction on different parties such as investors, businesses, and governments. Accurate forecasts affect investment choices, strategies for managing risks, long-term planning, fundraising endeavors, economic indicators, investor trust, and advancements in financial analysis technologies. Through creating strong predictive models, the project offers practical insights and aid in making well-informed decisions within the ever-changing stock market environment.

1.3 Problem Statement

The existing Stock price prediction system has several limitations that can obstruct user involvement and accessibility.

1. The SVR model struggles with capturing patterns effectively.
2. CNN models face challenges in interpreting their learnings.
3. RNN has an exploding and vanishing gradient problem.

1.4 Project Objectives

1. To address the challenge of accurately predicting stock prices in a volatile and complex market environment.
2. To overcome the limitations of traditional forecasting methods in stock price prediction.

1.5 Scope of Project

This project had focused on developing and evaluating a model to predict stock prices using DL techniques. Sophisticated DL algorithms have been employed to analyze a large amount of historical stock market data. The primary DL technique that had been used in this project was the LSTM network. LSTM networks had been particularly effective at capturing complex patterns and dependencies within data over time, making them ideal for predicting stock prices. Extensive historical data on stock prices, including crucial information such as the opening price, closing price, highest price, lowest price, and trading volume for each day, had been collected. Once the data had been collected, it had been cleaned and preprocessed to ensure accuracy and consistency.

This step involved handling missing values, removing outliers, and normalizing the data to make it suitable for training the LSTM model. The LSTM model was then defined and built. This involved selecting the appropriate architecture for the model, including the number of LSTM layers and units in each layer. The model had been trained on the preprocessed data to learn the patterns and trends in the stock market. The training process involved adjusting the model's parameters to minimize the prediction error. Subsequently, the model had been evaluated to determine its accuracy and effectiveness. This involved testing the model on a separate set of data that it had not seen before and calculating performance metrics such as RMSE and MAPE. These metrics had helped to quantify how well the model could predict future stock prices. The power of DL techniques, particularly LSTM networks, in analyzing and predicting stock prices has been demonstrated through this project. The importance of data preprocessing, model training, evaluation, and optimization in developing an effective prediction model has been highlighted. By following these steps, valuable insights into stock market trends and improved decision-making in financial investments have been aimed to be provided. It is essential to acknowledge the limitations of this project, which may include: the accuracy and completeness of historical market data significantly influencing the analysis, highlighting a dependency on data quality and availability. However, these factors can vary, leading to potential inconsistencies and inaccuracies in our models. Accurately predicting stock prices poses challenges due to the inherent volatility and unpredictability of financial markets. The dynamic nature of market conditions introduces complexities that can impact the reliability of our predictions, requiring continuous refinement and adaptation of our models to enhance their robustness and effectiveness.

1.6 Potential Project Applications

1. **Financial Decision Making:** The LSTM-based stock price prediction system can empower investors, traders, and financial analysts to make informed decisions regarding stock investments, portfolio management, and risk mitigation strategies.

2. **Algorithmic Trading:** Financial institutions and hedge funds can utilize the LSTM system for algorithmic trading, leveraging its predictive capabilities to automate trading decisions and execute transactions in real-time.

3. **Risk Assessment:** The LSTM model's ability to capture temporal dependencies can aid in risk assessment by providing insights into potential market fluctuations and identifying risk factors that may impact investment portfolios.

4. **Market Trend Analysis:** The LSTM system can analyze historical data to identify patterns and trends in stock prices, enabling stakeholders to anticipate market movements and capitalize on emerging opportunities.

5. **Financial Planning:** Individuals and businesses can utilize LSTM-based predictions for financial planning purposes, such as retirement planning, wealth management, and investment strategy development.

6. **Educational Tools:** The LSTM-based stock price prediction system can serve as an educational tool for learning about financial markets, data analysis, and machine learning techniques in finance.

7. **Research and Development:** Researchers and academics can leverage the LSTM system to study stock market dynamics, develop new forecasting models, and contribute to the advancement of financial analytics.

1.7 Originality of Project

This work stands out for its use of cutting-edge machine learning techniques, particularly advanced deep learning methods like LSTM networks. These techniques are crucial for accurately forecasting stock prices by capturing complex patterns in sequential data.The main focus is on creating reliable predictive models that leverage historical market data and economic indicators. Our project models provide actionable insights and support informed decision-making in the ever-changing stock market environment, benefiting individual investors, financial institutions, and corporations alike.

1.8 Organisation of Project

The material in this project report is organised into seven chapters. Chapter 1 introduces the problem topic this work tries to address, Chapter 2 contains the literature review of vital and relevant publications, pointing toward a notable research gap. Chapter 3 describes the methodology for the implementation of this project. Chapter 4 provides an overview of what has been accomplished. Chapter 5 contains some crucial discussions on the used model and methods. Chapter 6 mentions pathways for future research direction for the same problem or in the same domain. Chapter 7 concludes the project shortly, mentioning the accomplishment and comparing it with the main objectives.

2 LITERATURE REVIEW

2.1 Literature Review I:

The stock market has offered investors opportunities for profit but also involved risks. Various studies and theories regarding stock market timing strategies had been developed by researchers and expert investors. This study had examined the use of regression algorithms as a predictive tool for stock price patterns, utilizing a dataset that had been created through fundamental analysis. The dataset's features had consisted of statistical ratios, and all numerical data had been converted to numerical or numbered values. Predictive analytics using regression-based classifiers from WEKA had been applied to test ordinary data, and the results had been evaluated and compared. Since the inception of stock markets, extensive research had aimed to develop models for predicting stock price movements. Two prominent models had been the EMH and the Random Walk Theory. Eugene Fama's dissertation on the EMH in the 1960s had argued that stock prices reflected all available information in an active market, making it impossible to consistently outperform the market. This concept had aligned with Louis Bachelier's Random Walk Hypothesis, which posited that stock market prices evolved randomly and were unpredictable[1]. Professional investors had typically favored two primary approaches: fundamental analysis and technical analysis. Fundamental analysis had involved examining financial reports such as balance sheets and profit and loss statements to identify promising stocks based on their intrinsic attributes. This method had also included contrarian strategies, which had exploited human emotional biases in market behavior[1]. Technical analysis, in contrast, had identified chart patterns based on historical share prices to predict future trends, assuming that public information did not provide a competitive trading advantage. Recent research on stock market prediction techniques had increasingly focused on technological approaches, particularly machine learning. Machine learning algorithms had analyzed relationships between indicators to make forecasts. Regression techniques, with a history dating back to the 19th century, had formed part of this approach. A personalized transformation procedure that had converted numerical data into ordinal values based on a ranking system had been shown to enhance regression methods[1]. Among various regression methods that had been studied using WEKA, SMO Regression had been identified as particularly effective. Transforming real numbers into categorical ordinal data had improved the performance of regression approaches, especially when less structured data had been converted into more structured ordinal form. Further research into the impact of transforming different data types in regression techniques could have provided valuable insights for optimizing stock price prediction methods.

2.2 Literature Review II:

The utilization of neural networks in forecasting stock prices had garnered significant attention due to their unique capabilities. Neural networks had excelled in extracting context from complex or imprecise information, making them adept at identifying patterns and trends beyond human or conventional computational comprehension[2]. Their nonlinear nature had given them an edge over traditional linear models, and they could be retrained for new conditions while adjusting their weights as the framework evolved. This had made them particularly suitable for the highly nonlinear, dynamic, and ever-changing environment of the stock market. Extensive research and analysis had been required in both deep learning and financial theories, particularly regarding technical indicators, trading strategies, and stock market data analysis. Various applications of deep learning in financial domains had been explored, highlighting several advantages over traditional predictors, such as avoiding overfitting and handling input data correlations. The perfect neural network for specific research had remained elusive, requiring substantial study and analysis. Different types of neural networks had been compared, such as Support Vector Regression, Feed Forward Neural Networks, Convolutional Neural Networks, and Echo State Networks. Among these, LSTM networks had been traditionally successful for time-series prediction, with studies showing their effectiveness in learning patterns relevant for stock market prediction. Some research had indicated that MLP had outperformed LSTMs in predicting stock prices. The feature engineering process had involved a heuristic method, requiring a deep understanding of trading strategies and the basic concepts of the stock market. Neural networks had been shown to potentially outperform the Efficient Market Hypothesis by predicting market movements to some extent[2]. This had been demonstrated through comparative studies between LSTM models and MLP models, with the latter often performing better in short-term stock price prediction. Technical analysis had been required in guiding the framework to understand patterns from historical prices and predict future prices. The analysis of different neural network types and the selection of the most suitable one for specific research problems had been a significant challenge, often requiring extensive study and comparison. Future research had aimed to extend these models to real-time trading platforms, applying predictions to tick-by-tick data from stock markets. This had involved creating a comprehensive platform to execute trades based on these predictions in real-time, offering potential economic advantages by accurately predicting short-term stock price movements. Neural networks had presented a promising tool for forecasting stock prices[2]. Their ability to handle complex, nonlinear, and dynamic data had made them well-suited for the stock market, providing valuable insights and potentially outperforming traditional prediction models.

2.3 Literature Review III:

Stock price prediction had been a longstanding challenge in financial markets, with researchers and practitioners constantly seeking more accurate and reliable methods. Traditional approaches had utilized both linear models, such as ARIMA, and non-linear models, such as ARCH, to forecast stock prices based on historical data. However, these methods had often struggled to capture complex patterns and sudden market shifts, leading to limited predictive capabilities, especially in volatile markets. In recent years, the advent of deep learning algorithms, particularly RNN, LSTM, and CNN, had offered promising avenues for enhancing stock price prediction accuracy[3]. Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, and Soman K.P had collaborated on this domain in their paper titled ”Stock Price Prediction Using LSTM, RNN, and CNN - Sliding Window Model,” conducted at the Institution of Electrical and Electronics Engineers. Their work had focused on the application of deep learning algorithms to predict stock prices, emphasizing the importance of identifying and exploiting patterns, interactions, and hidden dynamics within time series data[3]. The authors had highlighted that while linear and non-linear models had been traditionally used for stock price prediction, their reliance on historical data alone had limited their ability to capture the underlying dynamics of the market[3]. In contrast, deep learning algorithms like RNN, LSTM, and CNN had been designed to learn from sequential data, enabling them to identify long-term dependencies, extract relevant features, and make more accurate predictions. The strength of employing RNN and LSTM had lied in their ability to analyze time-dependent data efficiently. These models had leveraged information from previous time steps to predict future instances, making them well-suited for capturing temporal dependencies and trends in stock prices. On the other hand, CNN, known for its prowess in image analysis, had been adapted to extract relevant features from financial data, aiding in predictive modeling by focusing on current input sequences and changes in trends[3]. One of the key findings of the study had been that CNN, by concentrating on current windows for prediction and analyzing changes in trends, had provided more accurate results compared to traditional methods. This had underscored the importance of utilizing advanced computational techniques like deep neural networks in stock price prediction, as they could capture latent dynamics, adapt to evolving market conditions, and offer insights into interrelations within the data. The performance of these models had been sensitive to hyperparameters and training techniques, requiring careful optimization and validation. While deep learning algorithms had excelled in capturing complex patterns and trends, they might still have struggled with extreme market events or anomalies that had deviated significantly from historical data patterns. Incorporating mechanisms for detecting and adapting to such events had remained an ongoing area of research within the field of financial forecasting[3]. The integration of deep learning algorithms, particularly RNN, LSTM, and CNN, had offered a promising avenue for enhancing stock price prediction accuracy. These models had demonstrated advanced computational capabilities in handling sequential data, extracting relevant features, and capturing hidden dynamics within financial time series.Further research and development had been needed to address challenges related to data availability, model robustness, and adaptation to unforeseen market shifts, ensuring the continued advancement of predictive modeling in financial markets. The interpretability of deep learning models had remained a concern, as understanding the underlying rationale behind predictions had been crucial in financial decision-making[3].

2.4 Literature Review IV:

The stock market had been a highly complex and dynamic system influenced by a wide range of factors such as economic conditions, company performance, political events, and global trends. Accurately predicting the future prices of stocks had been a challenging task with significant implications for investors, traders, financial institutions, and regulatory bodies. The study ”Stock Market Prediction Using LSTM Recurrent Neural Network” by Adil Moghar and Mhamed Hamiche, published by Elsevier B.V., had provided a thorough analysis of the effectiveness of LSTM networks in predicting stock prices. Learning models such as artificial neural networks, gradient-boosted regression trees, support vector machines, and random forests had revealed complex patterns characterized by non-linearity and relationships that had been difficult to detect with linear algorithms. These algorithms had also proven more effective and had handled multicollinearity better than linear regression models[4]. RNNs had been designed to detect patterns in sequential data, making them highly suitable for stock market analysis where data points had been temporally linked. Traditional RNNs had struggled with the vanishing gradient problem, which had hampered their ability to learn long-term dependencies. LSTMs had overcome this issue with their advanced architecture, which had included input, output, and forget gates. These gates had managed information flow, allowing LSTMs to effectively maintain and use information over extended periods, thereby boosting their predictive accuracy. In LSTM networks, each layer had built on the features extracted by the previous layer, enhancing the model's capability to identify complex patterns in stock price movements. These layers had been crucial for managing complex data and improving prediction accuracy. LSTM networks had significantly enhanced prediction accuracy by effectively capturing temporal dependencies in stock prices[4]. LSTMs had been able to track the progression of opening prices for a portfolio of assets, providing robust frameworks for predicting future trends. They had helped in portfolio management and risk assessment as well. LSTMs had modeled long-term dependencies and trends, essential for reliable stock market predictions, by maintaining cell states over time and selectively forgetting or retaining information. LSTMs had been proficient at analyzing time-dependent data, retaining crucial information for future use, and adapting to changing trends, making them highly suitable for financial forecasting. The independence of LSTM cells had allowed the model to process data efficiently without the need for filters to transfer values from one cell to another. This independence had improved the model's ability to manage and use past data for predicting future trends. LSTMs had utilized earlier data stages to learn patterns and predict future movements, which had been vital for stock market analysis. LSTM networks had required more epochs and had been slower to train compared to simpler models. The training process for LSTMs had been computationally intensive and time-consuming, which could have hindered their practical application in real-time forecasting[4].

2.5 Literature Review V:

The application of LSTM for short-term stock price prediction had been extensively explored and had shown significant improvements over traditional methods. Traditional approaches, such as Linear Regression and SVR, had been employed to predict stock prices but had not achieved satisfactory accuracy[5]. ARIMA models had also been used, but their performance had been limited due to the high variability and non-linear nature of stock prices. LSTM networks had gained traction due to their proven effectiveness in various analytical fields. ANN had initially been used to handle non-linear data, but their performance had been insufficient for time-series predictions. RNN had been introduced to improve accuracy, but they had faced limitations, such as the inability to store long-term dependencies and the vanishing gradient problem. To address these issues, LSTM networks had been applied, offering better performance by retaining long-term dependencies and mitigating the vanishing gradient problem through mechanisms like input, forget, and output gates[5]. LSTM networks, being a type of RNN, had been particularly suitable for time-series data due to their ability to capture temporal dependencies. The architecture of LSTM had included input, forget, and output gates, which had helped manage the flow of information and retain relevant data across time steps. These features had enabled LSTM to effectively model the complex patterns and dependencies in stock price data. SVR, a variant of SVM, had aimed to minimize the error between predicted and actual values by finding the most suitable hyperplane. SVR had used different kernel functions, such as linear, sigmoid, and polynomial, to transform the data and improve prediction accuracy. Despite its effectiveness in some cases, SVR had been outperformed by LSTM in terms of accuracy for stock price prediction. LSTM had outperformed SVR across various stock indices[5]. LSTM had achieved lower MAPE values compared to SVR, indicating better prediction accuracy. The results had demonstrated that LSTM could handle the non-linear and volatile nature of stock prices more effectively than SVR. The application of LSTM and CNN on stock indexes like NYSE and NSE had shown that neural network-based approaches outperformed traditional linear models like ARIMA. Deep convolutional networks combined with candlestick charts had achieved high accuracy in stock market predictions. These studies had underscored the potential of deep learning techniques in capturing the intricate patterns in stock price data. The research had also highlighted the importance of selecting appropriate evaluation metrics and data preprocessing techniques[5]. Metrics like MAPE, MAE, and RMSE had been commonly used to assess model performance. Proper data preprocessing, including feature selection and normalization, had been crucial for improving the accuracy of predictions. LSTM's ability to retain long-term dependencies and handle non-linear data had made it a superior choice for stock price prediction. The research had validated these findings through extensive experiments on various stock indices, confirming that LSTM had provided better prediction accuracy and robustness[5].

3 METHODOLOGY

3.1 Theoretical Formulations

1. **Traditional Time Series Analysis**

   ARIMA : ARIMA models capture linear relationships in time series data through the combination of three elements:

   AR : Autoregression uses past values of the variable to predict future values.

   Integrated: Integration involves differencing the observations to achieve stationarity in the time series.

   MA : Moving Average uses past forecast errors to model the time series in a regression-like approach.

2. **Machine Learning Models**

   SVM : Support Vector Machines (SVMs) are supervised learning algorithms employed for both classification and regression tasks. In the context of regression, SVMs identify the hyperplane that best fits the data by maximizing the margin between the data points and the hyperplane, ensuring that the prediction error is minimized within a specified threshold.

3. **Deep Learning Models**

   LSTM : LSTM networks are a type of RNN specifically designed to capture long-term dependencies in sequential data. LSTMs utilize internal memory cells and gating mechanisms (input, forget, and output gates) to regulate the flow of information, enabling effective learning of temporal patterns.

3.2 Mathematical Modelling

When using an LSTM network for stock price prediction, the mathematical modeling involves adapting the standard LSTM equations to the specific task of predicting stock prices.

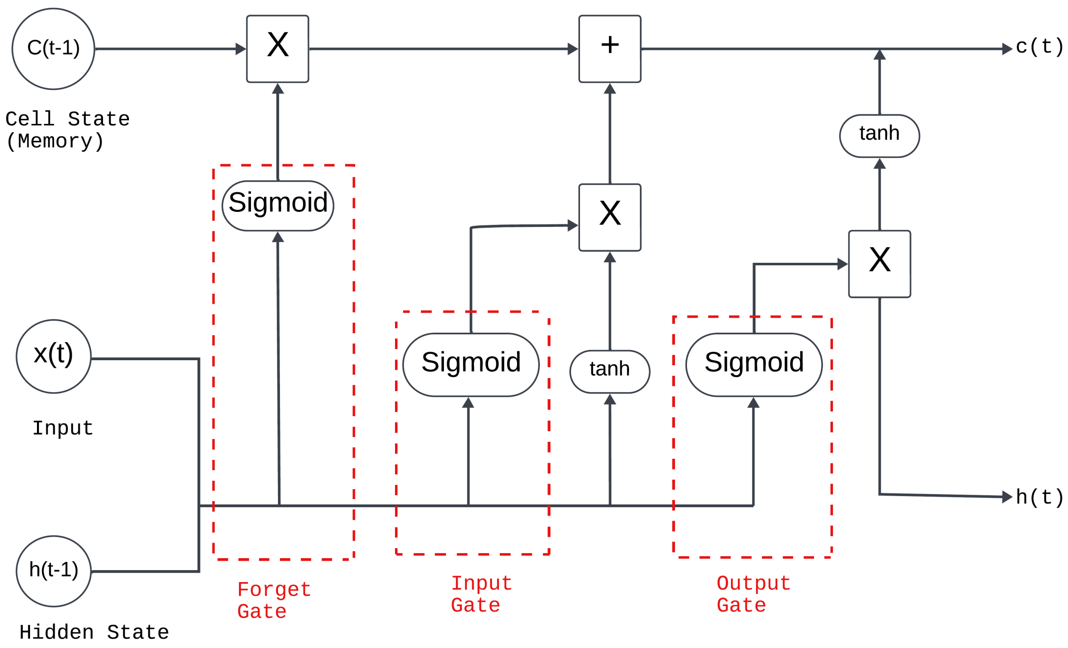

Figure 3.1: LSTM Architecture

Here's a conceptual breakdown of how the LSTM equations can be applied in the context of stock price prediction:

Let's denote:

x_t as the input features (such as historical stock prices) at time t.

h_{t-1} as the hidden state at time $t-1$.

C_{t-1} as the cell state at time $t-1$.

y_t as the predicted stock price at time t.

Forget Gate:

The information that is no longer useful in the cell state is removed with the forget gate. Two inputs x_t (input at the particular time) and h_{t-1} (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state, the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.1}$$

Input Gate:

The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs h_{t-1} and x_t. Then, a vector is created using the tanh function that gives an output from -1 to +1, which contains all the possible values from h_{t-1} and x_t. At last, the values of the vector and the regulated values are multiplied to obtain useful information.

$$f_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.2}$$

Output Gate:

The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying the tanh function on the cell. Then, the information is regulated using the sigmoid function and filtered by the values to be remembered using inputs h_{t-1} and x_t. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.3}$$

Candidate Cell State:

The candidate cell state represents a potential update to the cell state, which is proposed based on the current input and the previous hidden state. This candidate cell state is calculated using a combination of the input and the hidden state through a set of weights, followed by, the hyperbolic tanh activation function.

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3.4}$$

Cell State Update:

The LSTM also maintains a cell state vector C_t, which is responsible for storing long-term information over the course of the sequence. The cell state is initialized to a vector of zeros at the beginning of the sequence.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{3.5}$$

Hidden State:

The LSTM maintains a hidden state vector, h_t, which represents the current "memory" of the network. The hidden state is initialized to a vector of zeros at the beginning of the sequence.

$$h_t = o_t \cdot \tanh(C_t) \tag{3.6}$$

Output Layer:

$$y_t = W_y \cdot h_t + b_y \tag{3.7}$$

Where:

i_t, f_t, o_t are the input, forget, and output gates' outputs respectively.

$\tilde{C}_t$ is the candidate cell state.

C_t is the cell state at time t.

h_t is the hidden state at time t.

W_i, W_f, W_o, W_c, W_y are weight matrices.

b_i, b_f, b_o, b_c, b_y are bias vectors.

σ is the sigmoid activation function.

$\tanh$ is the hyperbolic tangent activation function.

This mathematical modeling allows the LSTM network to learn temporal dependencies in the historical stock price data and generate predictions for future stock prices based on the input features. Adjustments to the architecture and hyperparameters can be made through experimentation to improve prediction accuracy.

Activation Function:

The activation functions are crucial in LSTM networks for regulating information flow, maintaining and updating the cell state, and generating the final output of the LSTM cell.

1. **Sigmoid Function (σ):** This is used in the LSTM gates (input gate, forget gate, output gate) to control the flow of information.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3.8}$$

Purpose: It squashes input values between 0 and 1, allowing the gates to decide which information to let through (0 for forget, 1 for retain or input).

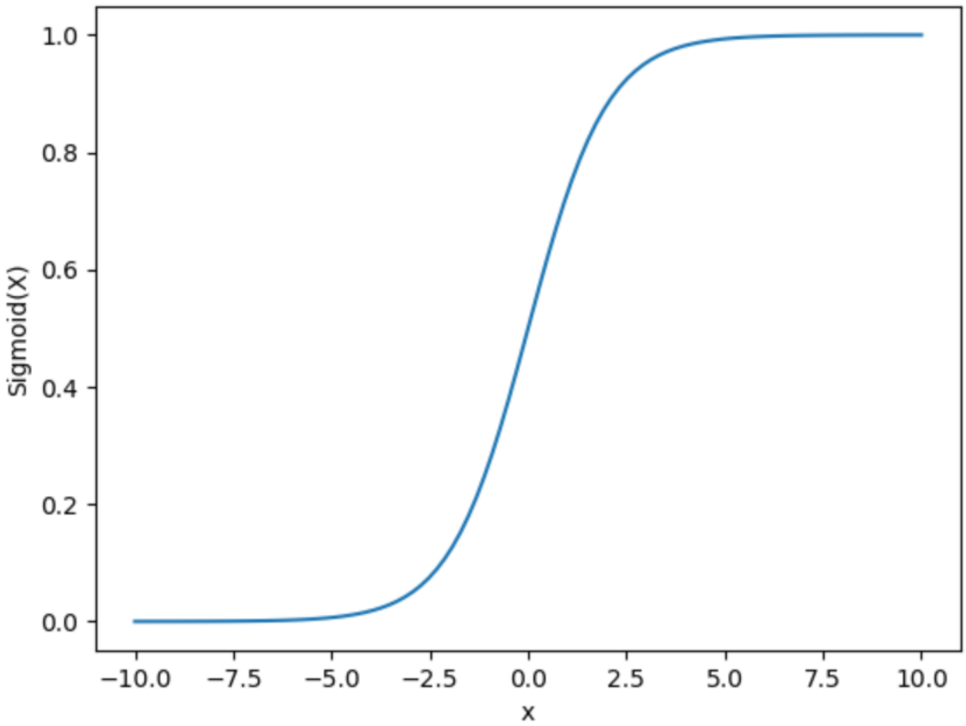

Figure 3.2: Sigmoid Function

2. **Hyperbolic Tangent (Tanh) Function:** This function is used to regulate the cell state updates and the output of the LSTM cell.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.9}$$

Purpose: It squashes input values between -1 and 1, controlling the updating of the cell state and the output range of the cell.

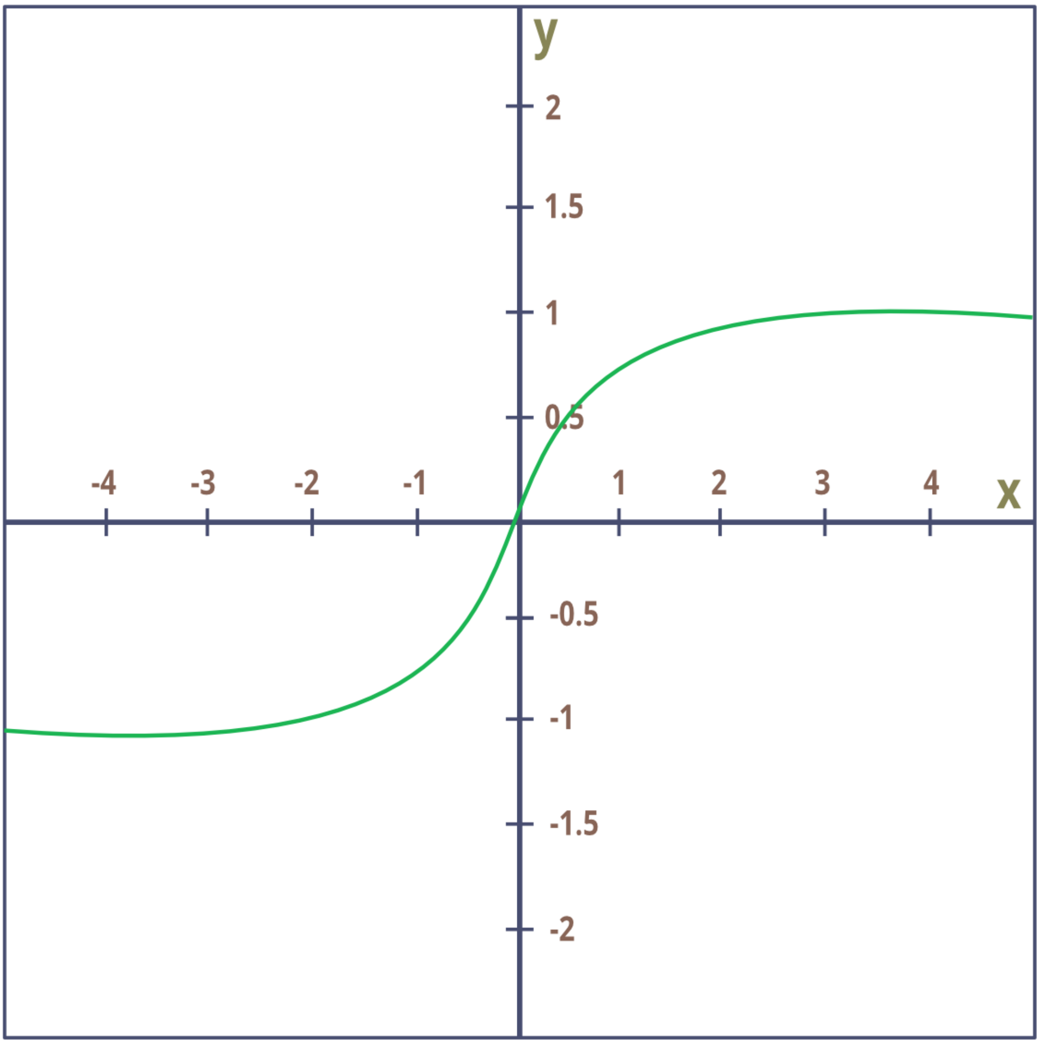

Figure 3.3: Hyperbolic Tangent function

3.3 System Block Diagram

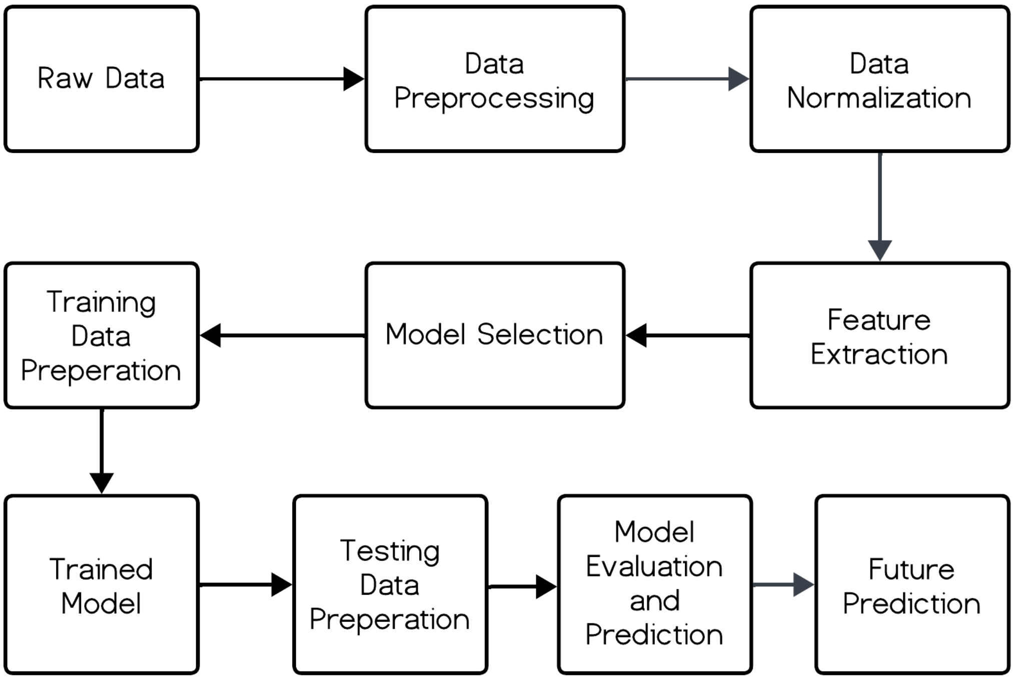

Figure 3.4: System Block Diagram

Raw Data: The necessary information had been collected, including the date, security ID, and closing value of NABIL and CBBL through NEPSE office. Date, closing value, opening value, high and low value, quantity and turnover had been collected of SBL through the MEROLAGANI website.

Data Preprocessing: The raw data had been preprocessed to ensure quality and consistency. This preprocessing had involved several steps:

- **Missing Values:** The missing values in the dataset had been removed.
- **Consistency:** The data had been checked for consistency, ensuring that all records followed the same structure and format.

Data Normalization: After preprocessing, the data had been normalized. Normalization had scaled the data to a standard range, typically between 0 and 1. This step had been crucial because DL models perform better when the input data is on a similar scale, preventing features with larger ranges from disproportionately influencing the model.

Feature Extraction: The relevant features had been extracted from the normalized data. The last 60 days closing prices had been selected as input features for the model.

Model Selection: The appropriate model had been selected for this project. In this context, an LSTM model, known for its effectiveness in handling time series data, had been chosen. LSTM models are capable of learning long-term dependencies and patterns in sequential data, making them suitable for stock price prediction.

Training Data Preperation: The dataset had been divided into training and testing sets. Seventy-five percent of the data had been used for training. This training process had involved feeding the model with input data and corresponding output labels (actual stock prices) so it could learn the underlying patterns and relationships. Training data had been prepared, consisting of sequences of historical stock prices and corresponding dates. The training data had been created by splitting the dataset into sequences of 60 days.

Trained Model: The LSTM model had been trained on the training data. During training, the model parameters had been adjusted to minimize prediction errors. Techniques such as backpropagation and gradient descent had been used to optimize the model's performance. The model had been compiled using the Adam optimizer and mean squared error loss function. The data had been trained with a batch size of 64 and 100 epochs.

Testing Data Preperation: The remaining twenty five percent of the dataset had been reserved as testing data. This testing data had been kept separate from the training data to provide an unbiased evaluation of the model's performance.

Model Evaluation and Prediction: The trained LSTM model had been evaluated using the testing data. The model's predictions had been compared to the actual stock prices to assess its accuracy and performance.

Future Prediction: When the model's performance had been deemed satisfactory, it had been used to predict future stock prices based on new input data. The final output of the model had been the predicted stock prices for the next 30 days.

3.4 Instrumentation Requirements

Requirements	Details
• Software Requirements	• Operating system: Windows 10 or 11 • Browser: Edge preferred (any) • Application: VS Code, Jupyter Notebook
• Hardware Requirements	• RAM: 4 GB minimum • Internet: Minimum 10 Mbps • SSD: 128 GB • CPU: i5, at least 2.4GHz
• Language	• Python
• Libraries Requirements	• Sklearn • Pandas • Matplotlib • Math • Keras • Numpy

Table 3.1: Instrumentation Requirements

3.5 Dataset Explanation

The dataset used in this project was sourced from two primary sources: NEPSE and Merolagani, a leading financial portal in Nepal. NEPSE provided historical stock market data for companies listed on the Nepal Stock Exchange, including detailed information on stock prices such as the Date, Security ID and closing prices and other relevant financial indicators. Merolagani offered additional financial data and insights, serving as a comprehensive resource for stock market analysis in Nepal. By incorporating data from Merolagani, the dataset was enriched with more detailed financial metrics and insights, enhancing the ability to analyze and predict stock market behaviors. By combining data from these two sources, the dataset encompassed a wide range of stock market information, enabling comprehensive analysis and modeling for stock price prediction. The inclusion of various financial indicators provided a robust foundation for developing accurate predictive models. For the prediction models, it was essential to utilize the ”Close” data as a key feature. This is particularly important when employing LSTM networks, a type of RNN that is highly effective for sequence prediction tasks. LSTM networks are well-suited for stock price forecasting as they can learn from the historical sequences of closing prices, capturing temporal dependencies and trends in the data. The NABIL and CBBL stock data had been taken from NEPSE office, covering a time frame from 2007 to 2024. This dataset included the Date, Security ID, and Close value on a daily basis, providing a long-term view of the stock performance of NABIL and CBBL.

Date	Security_id	Close
8/15/2007	NABIL	5113.75
8/16/2007	NABIL	5010
8/19/2007	NABIL	5250
8/20/2007	NABIL	5320
8/21/2007	NABIL	5400
8/22/2007	NABIL	5390
8/23/2007	NABIL	5390
8/24/2007	NABIL	5350
8/26/2007	NABIL	5230
8/27/2007	NABIL	5350
8/27/2007	NABIL	5350

Figure 3.5: Dataset of NABIL stock

The SBL stock data had been taken from Merolagani, covering a timeframe from 2014 to 2024. This dataset included more detailed daily records such as the date, close value, the highest and lowest prices of the day, the opening price, the quantity of shares traded, and the turnover. This comprehensive dataset allowed for a detailed analysis of SBL stock performance over the specified period. The combined dataset provided a rich and detailed view of the stock market in Nepal, supporting advanced predictive modeling and analysis for stock price forecasting.

Date	Close	High	Low	Open	Qty	Turnover
7/30/2014	883	895	865	870	13794	9133539
8/1/2014	870	887	870	881	5964	9133539
8/2/2014	881	900	880	865	10172	9133539
8/3/2014	851	870	850	860	11684	9133539
8/4/2014	835	851	825	851	10907	9133539
8/4/2014	860	883	860	883	11274	9133539
8/5/2014	837	864	836	835	7406	6257139
8/6/2014	870	899	853	837	46306	40362840
8/7/2014	877	884	850	870	35034	29077155
8/12/2014	870	874	855	877	19156	16550583
8/13/2014	882	910	850	870	20410	17817317
8/14/2014	885	900	880	882	14962	13264123

Figure 3.6: Dataset of SBL stock

The SBL dataset has 2124 rows and 7 columns.

Column details:

Date: Object type (string) representing the date of each record.

Close, High, Low, Open, Turnover: Float type representing various stock prices and financial metrics.

Qty: Integer type representing the quantity of shares traded.

```
<class 'pandas.core.frame.DataFrame'>
Index: 2101 entries, 20 to 2120
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Date        2101 non-null   object
 1   Close       2101 non-null   float64
 2   High        2101 non-null   float64
 3   Low         2101 non-null   float64
 4   Open        2101 non-null   float64
 5   Qty         2101 non-null   float64
 6   Turnover    2101 non-null   float64
 7   MA7         2101 non-null   float64
 8   MA21        2101 non-null   float64
 9   Volatility  2101 non-null   float64
dtypes: float64(9), object(1)
memory usage: 180.6+ KB
```

Figure 3.7: SBL Data Info

	Close	High	Low	Open	Qty	Turnover	MA7	MA21	Volatility
count	2101.000000	2101.000000	2101.000000	2101.000000	2101.000000	2.101000e+03	2101.000000	2101.000000	2101.000000
mean	441.412280	447.988063	435.354831	442.290205	45781.879581	1.998450e+07	442.166859	444.122635	10.529914
std	226.550934	229.841720	222.699672	226.129740	69937.658277	3.543659e+07	226.289836	225.149553	16.121762
min	215.000000	216.400000	212.200000	214.100000	45.000000	1.395000e+04	219.057143	220.676190	0.566947
25%	298.000000	302.000000	295.000000	299.000000	10016.000000	3.878339e+06	299.000000	300.714286	3.447912
50%	342.000000	345.000000	337.400000	343.000000	22753.000000	7.960330e+06	341.142857	338.714286	6.078847
75%	511.000000	521.000000	504.200000	514.000000	49809.000000	1.922214e+07	512.285714	525.419048	11.060440
max	1416.000000	1448.000000	1400.000000	1440.000000	901446.000000	4.262128e+08	1391.285714	1346.952381	185.419088

Figure 3.8: SBL Data Analysis with Technical Indicators

Statistical Summary

Close (Closing Price)

- **Count:** 2101
- **Mean:** 441.41
- **Standard Deviation:** 226.55
- **Minimum:** 215.00
- **25%:** 298.00
- **50% (Median):** 342.00
- **75%:** 511.00
- **Maximum:** 1416.00

The closing price ranges significantly, with a minimum of 215 and a maximum of 1416, indicating considerable volatility. The 25^{th} percentile is 298, the median is 342.00, and the 75^{th} percentile is 511.00, showing the distribution of the closing prices.

High (Highest Price of the Day)

- **Count:** 2101
- **Mean:** 447.98
- **Standard Deviation:** 229.71
- **Minimum:** 216.40
- **25%:** 302.00
- **50% (Median):** 345.00
- **75%:** 521.00
- **Maximum:** 1448.00

The highest prices have a slightly higher mean and maximum value compared to closing prices, as expected, since the highest price of the day often exceeds the closing price. The 25th percentile is 302, the median is 345, and the 75th percentile is 521.00.

Low (Lowest Price of the Day)

- **Count:** 2101
- **Mean:** 435.35
- **Standard Deviation:** 222.69
- **Minimum:** 212.20
- **25%:** 295.00
- **50% (Median):** 337.40
- **75%:** 504.20
- **Maximum:** 1400.00

The lowest prices of the day also show significant variation, with the mean close to the closing price mean. The 25th percentile is 295, the median is 337.40, and the 75th percentile is 504.20.

Open (Opening Price)

- **Count:** 2101
- **Mean:** 442.29
- **Standard Deviation:** 226.12
- **Minimum:** 214.10
- **25%:** 299.00
- **50% (Median):** 343.00
- **75%:** 514.00

- **Maximum:** 1440.00

The opening prices are similar in range to the closing prices, with a comparable mean and standard deviation. The 25th percentile is 299, the median is 343.00, and the 75th percentile is 514.00.

Qty (Quantity of Shares Traded)

- **Count:** 2101
- **Mean:** 45781.87
- **Standard Deviation:** 69937.65
- **Minimum:** 45.00
- **25%:** 10016.00
- **50% (Median):** 22753.00
- **75%:** 49809.00
- **Maximum:** 901446.00

There is a high degree of variability in the number of shares traded daily, indicating periods of high and low trading activity. The 25th percentile is 10016.00, the median is 22753.00, and the 75th percentile is 49809.00.

Turnover

- **Count:** 2101
- **Mean:** 19,984,500.00
- **Standard Deviation:** 35,436,590.00
- **Minimum:** 13,950.00
- **25%:** 3,878,339.00

- **50% (Median):** 7,960,330.00
- **75%:** 19,222,140.00
- **Maximum:** 426,212,800.00

Turnover shows extremely high variability, reflecting both the quantity of shares traded and their prices. The 25th percentile is 3,878,339 the median is 7,960,330, and the 75th percentile is 19,222,140.

MA7 (7-day Moving Average)

- **Count:** 2101
- **Mean:** 442.16
- **Standard Deviation:** 226.28
- **Minimum:** 219.05
- **25%:** 299.00
- **50% (Median):** 341.14
- **75%:** 512.28
- **Maximum:** 1391.28

The 7-day moving average smooths out daily fluctuations, providing a short-term trend indication. The 25th percentile is 299.00, the median is 341.14, and the 75th percentile is 512.28.

MA21 (21-day Moving Average)

- **Count:** 2101
- **Mean:** 444.12
- **Standard Deviation:** 225.14

- **Minimum:** 220.67
- **25%:** 300.71
- **50% (Median):** 338.71
- **75%:** 525.41
- **Maximum:** 1346.95

The 21-day moving average provides a longer-term trend indication compared to the 7-day moving average. The 25th percentile is 300.71, the median is 338.71, and the 75th percentile is 525.41.

Volatility

- **Count:** 2101
- **Mean:** 10.52
- **Standard Deviation:** 16.12
- **Minimum:** 0.56
- **25%:** 3.44
- **50% (Median):** 6.07
- **75%:** 11.06
- **Maximum:** 185.41

Volatility measures the degree of variation in stock prices, with higher values indicating more significant price fluctuations. The 25thh percentile is 3.44, the median is 6.07, and the 75th percentile is 11.06.

3.6 Description of Algorithms

The LSTM Gates and Operations are:

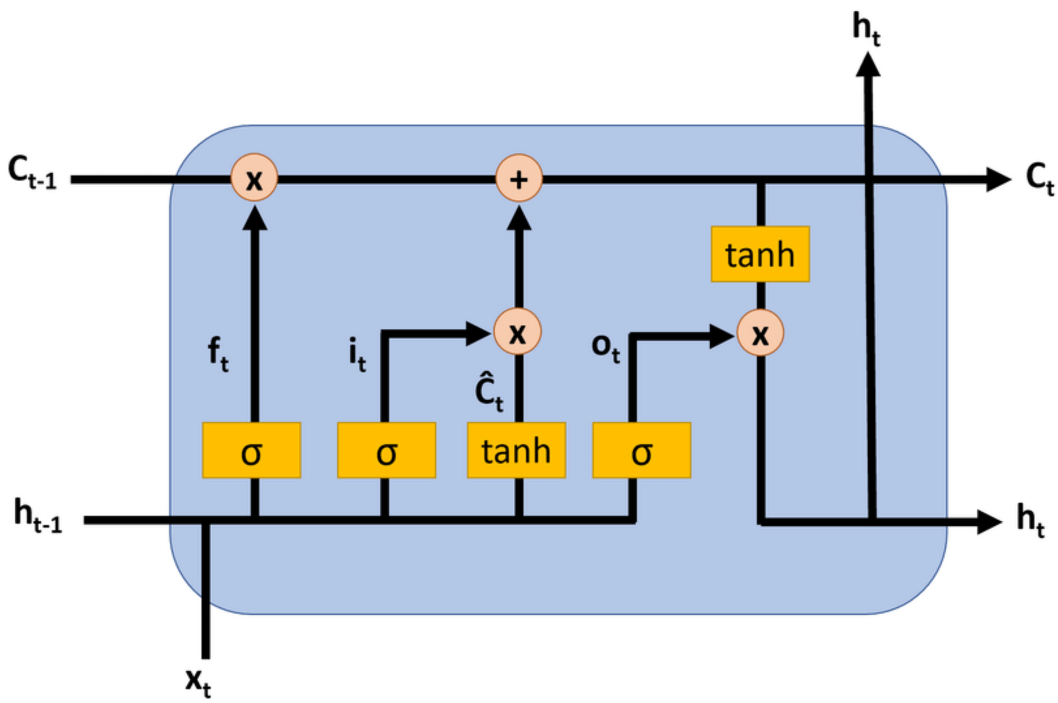

Figure 3.9: LSTM CELL

Figure 3.9 depicts the comprehensive structure of an LSTM cell, which includes three key gates:

1. Forget gate
2. Input gate
3. Output gate

1. **Forget Gate Function:** Determines which information from the previous time step should be discarded.

   Operation: Utilizes a sigmoid activation function on a combination of the prior hidden state and current input, producing values between 0 and 1 for each element in the memory cell.

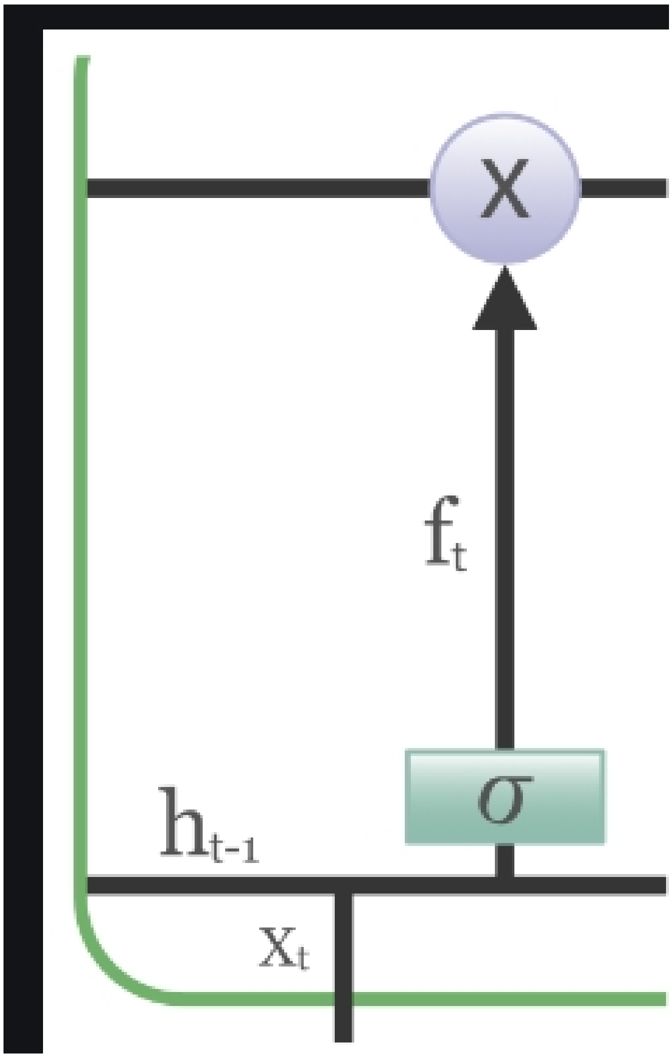

Figure 3.10: Forget Gate

Consider an example in a text prediction scenario. Suppose the LSTM is processing the following sentence:

Anu is a nice person. Denish on the other hand is evil.

We have two sentences that include a full stop (.). The first sentence is, “Anu is a nice person”, while the second sentence is, “Denish on the other hand is evil”. The first sentence mentions “Anu”. However, when we see the full stop, we start talking about “Denish”. When we switch from the first sentence to the second, our network must understand that we are no longer discussing “Anu”. The subject is now “Denish”. The Forget gate in the LSTM network allows it to forget about “Anu”.

The forget gate removes unnecessary information from the cell state, optimizing the LSTM network’s performance by discarding less important details. The gate receives two inputs: h_{t-1} (the previous cell's hidden state) and x_t (the current input). These inputs are multiplied by weight matrices and added to a bias before being passed through a sigmoid function. The sigmoid function outputs a vector with values between 0 and 1, corresponding to each element in the cell state. A '0' indicates the information should be forgotten, while a '1' signifies it should be retained. This vector is then multiplied with the cell state.

2. **Input Gate**

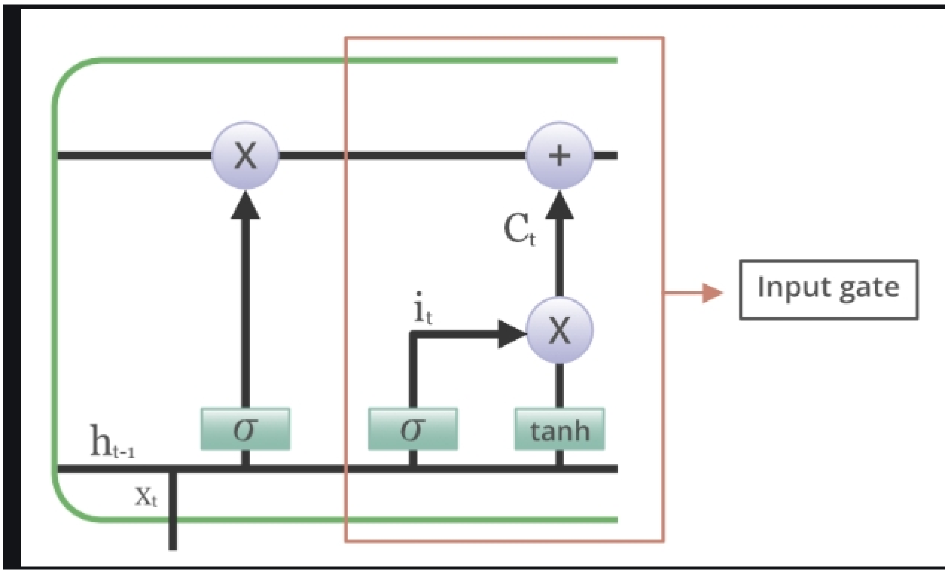

Figure 3.11: Input Gate

Function: Determines which new information should be stored in the memory cell.
Operation: Combines the current input and the previous hidden state, passing the result through a sigmoid activation to produce a candidate value for updating the memory cell. This candidate value is then scaled by another sigmoid function.

Consider another example where the LSTM processes the sentence:
Denish knows swimming. He told me over the phone that he had served in the navy for four long years.
In both sentences, we discuss "Denish". However, each sentence provides different information about "Denish". According to the first sentence, he is familiar with swimming. The second sentence informs us that he used the phone and served in the navy for four years. Which information in the second sentence is relevant

to the first sentence? Is it because he used a phone, or because he served in the Navy? In this case, it makes no difference whether he told us over the phone or in another way. The fact that this was conveyed over the phone is less significant and can be ignored and the important information is that he served in the Navy. This is the information we want our model to remember for future use. This is the Input gate's function and manages the inclusion of such relevant information into the cell state.

The input gate's function involves a three-step process:

(a) A sigmoid function filters the information from h_{t-1} and x_t to decide which values should be added to the cell state.

(b) A tanh function generates a vector of potential values to add, ranging from -1 to +1.

(c) The regulatory filter's (sigmoid gate's) output is multiplied with the tanh vector, and this result is added to the cell state.

This ensures only significant and non-redundant information is added to the cell state.

3. **Output Gate**

Function: Regulates the flow of information from the memory cell to the network output.

Operation: Combines the current input with the previous hidden state, passes the result through a sigmoid activation function, and determines how much of the cell state's contents should be passed to the output.

Consider an example:

Denish fought single-handedly with the enemy and died for his country. For his contribution brave __________.

In this task, we must complete the second sentence. When we see the word "brave", we know we are referring to a person. In the sentence, only "Denish" is brave. We can't say whether the enemy or the country are brave. Based on this expectation, we must find the appropriate word to fill in the blank. That word is our output, and it is the responsibility of the Output gate.

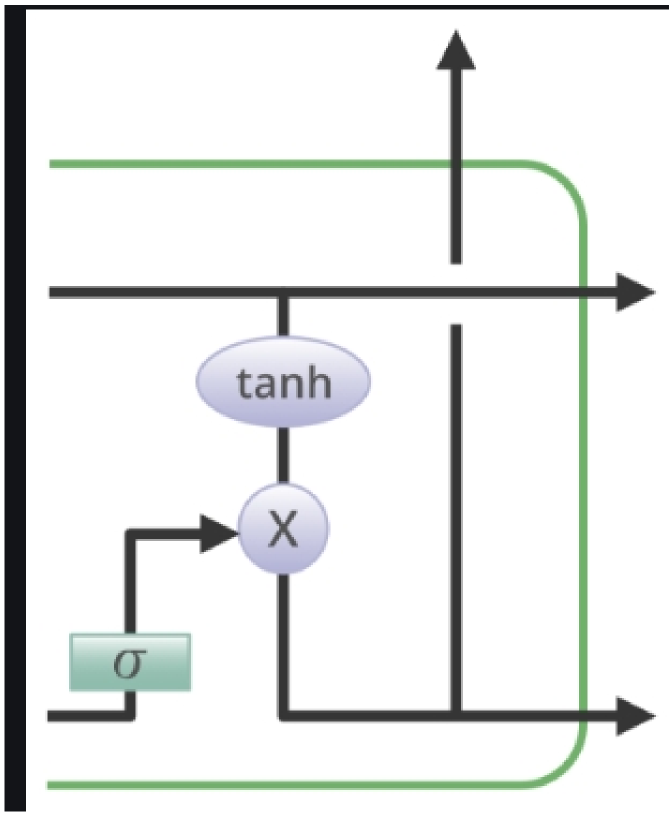

Figure 3.12: Output Gate

The output gate operates in three steps:

(a) A tanh function is applied to the cell state, scaling the values to the range -1 to +1.

(b) A filter is created using h_{t-1} and x_t to determine which values from the tanh vector should be output. This filter employs a sigmoid function.

(c) The regulatory filter's output is multiplied with the tanh vector, and the result is sent as the output and also to the next cell's hidden state.

In the example, the filter ensures that "Bob" is selected as the output, based on the input and hidden state values applied to the cell state vector.

3.7 Elaboration of Working Principle

LSTM networks had emerged as a popular choice for stock price prediction due to their ability to effectively capture and model temporal dependencies in sequential data. Here's a detailed elaboration of how LSTM worked in the context of stock price prediction:

Data Preprocessing

- **Raw Input Data Preprocessing Handling Missing Values** Missing values in the dataset had been identified and either filled using appropriate imputation techniques or removed.

- **Normalization** The data had been normalized to bring all features to a similar scale, improving the model's performance. Min-max scaling had been applied, transforming the data to a range of [0, 1]:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.10}$$

  where x is the original feature value, and x' is the normalized value.

- **Data Splitting** The dataset had been split into training and testing sets, typically using an 80-20 or 70-30 split, to evaluate the model's performance on unseen data.

Deep Learning Ready Data Manipulation

- **Sequence Creation:**
  For the LSTM model, the data had been structured into sequences. Each sequence represented a series of past stock prices used to predict the next stock price. For example, if the sequence length was set to 60, each input sequence consisted of 60 days of past prices.

- **Reshaping:**
  The data had been reshaped into a 3D array of shape (number of sequences, sequence length, number of features), which is the required input shape for LSTM networks.

Example Calculation: Assuming a sequence length of 60 and one feature (closing price), the data had been reshaped as follows:

$$\text{Input Shape} = (n - 60, 60, 1) \tag{3.11}$$

where n is the total number of data points.

Model Architecture and Training

1. **LSTM Layers**

   The Long Short-Term Memory (LSTM) model had been designed with multiple LSTM layers to capture the temporal dependencies present in the stock price data. Each LSTM layer contained memory cells that stored and processed information from previous time steps, which allowed the model to learn patterns over time. The LSTM layers were equipped with three types of gates: input, forget, and output gates. These gates regulated the flow of information, ensuring that only relevant data were retained and used for future predictions.

   (a) **Input Gate**

   The input gate managed the addition of new information to the cell state. For the stock price prediction model, it combined the current stock price input with the previous hidden state from the last time step. A sigmoid activation function was applied to this combination, resulting in a value between 0 and 1. This value determined the extent to which the new information influenced the cell state. For example, if the sigmoid function output was close to 1, a significant portion of the new stock price data was added to the cell state, allowing the model to adapt to new market conditions.

   (b) **Forget Gate**

   The forget gate regulated which information from the previous time step should be retained or discarded from the cell state. By applying a sigmoid activation function to the current stock price input and the previous hidden state, it produced a value between 0 and 1 for each element in the cell state. A value of 1 indicated full retention of previous information, while a value of 0 indicated complete forgetting. This mechanism allowed the model to decide whether to keep historical stock price patterns or to discard them based on their relevance to current predictions.

(c) **Output Gate**

The output gate controlled the flow of information from the cell state to the hidden state and the final network output. It applied a sigmoid activation function to the current stock price input and the previous hidden state, resulting in a value that determined how much of the cell state should be outputted. This regulated output was then passed through a tanh activation function to produce the next hidden state, which encapsulated the relevant information needed for predicting the next stock price. This gate ensured that the model's predictions were based on a filtered and refined version of the cell state, enhancing the accuracy of the stock price predictions.

(d) **Dense Layer**

At the end of the LSTM layers, a fully connected dense layer had been added to produce the final predicted stock price. The dense layer combined the outputs of the LSTM layers and applied a linear transformation to generate the prediction.

LSTM Layer 1

- **Type:** LSTM
- **Units:** 50
- **Return Sequences:** True
- **Input Shape:** (x_train.shape[1], 1)
- **Activation Function:** `tanh`
- **Purpose:** This layer processes the input data and returns the full sequence of outputs. Returning the full sequence (`return_sequences=True`) allows the next LSTM layer to receive a sequence as its input, helping the model capture more complex temporal dependencies.

LSTM Layer 2

- **Type:** LSTM
- **Units:** 50
- **Return Sequences:** False
- **Activation Function:** `tanh`
- **Purpose:** This layer takes the sequence of outputs from the previous LSTM layer and returns only the last output in the sequence. This is typically done before passing the data to the dense layers, where only the final output is of interest for further processing.

Dense Layer 1

- **Type:** Dense
- **Units:** 25
- **Activation Function:** linear
- **Purpose:** This fully connected layer processes the output from the LSTM layers. It applies weights and biases to the input and passes it through an activation function to create a more complex representation of the data.

Dense Layer 2

- **Type:** Dense
- **Units:** 1
- **Activation Function:** linear
- **Purpose:** The final dense layer with a single unit produces the final output of the model, which is a single predicted value (e.g., the predicted stock price).

2. **Training**

   The model had been trained using the training dataset. The training process involved forward and backward propagation to minimize the loss function, which was Mean Squared Error (MSE) in this case. The Adam optimizer had been used to update the model's weights and biases during training. The Adam optimizer is known for its efficiency in handling sparse gradients and adaptive learning rates, which helped in speeding up the convergence of the model.

Post-Processing of Model Output

The model's output, which had been in the normalized scale, had been post-processed to convert it back to the original scale using the inverse of the normalization formula:

$$x = x' \times (\max(x) - \min(x)) + \min(x) \tag{3.12}$$

where x' is the predicted normalized value, and x is the denormalized predicted value.

Evaluation and Validation

The performance of the LSTM model had been evaluated using metrics such as MSE, MAE, or MAPE. The model's predictions had been compared to the actual stock prices in the validation or test dataset to assess its accuracy and generalization ability.

3.8 Verification and Validation

Verification

Verification had been conducted to ensure that the LSTM model for stock price prediction had been built correctly and met the specified requirements.

1. **Data Preprocessing Verification:** The stock market data had been preprocessed to handle missing values, normalize the data, and split it into training and test sets. Data transformation steps, such as scaling, had been verified to ensure they were correctly implemented.

2. **Model Architecture Verification:** The LSTM model architecture had been reviewed to confirm it matched the specified design, including the number of layers, units per layer, activation functions, and other hyperparameters. Compilation of the model had been checked for correctness, ensuring the appropriate loss function and optimizer had been used.

3. **Training Process Verification:** The training process had been implemented correctly, with verification that the model had been trained on the correct data, using the right batch size, number of epochs, and validation split.

4. **Code Review and Testing:** A code review had been conducted to identify any bugs or logical errors. A unit tests for individual components of the model, such as data preprocessing functions and model building functions, had been implemented to ensure correctness.

Validation

Validation had been performed to ensure the model's performance on data and that it met the performance criteria. The relevance of the chosen metrics had been discussed to judge the output effectively.

1. **Performance Metrics:**
   The model's performance had been evaluated using metrics such as MSE, R^2, and MAPE on the test dataset, ensuring generalization to unseen data.

2. **Definitions and Formulas of Metrics:**

Mean Squared Error: MSE is the average of the squares of the errors—that is, the average squared difference between the actual and predicted values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.13}$$

Mean Absolute Error : MAE is the average of the absolute differences between the actual and predicted values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.14}$$

R^2 **/ Coefficient of Determination**:

The Coefficient of Determination measures the proportion of the variance in the dependent variable that is predictable from the independent variables

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.15}$$

Root Mean Square Error: The differences between predicted and observed values. It is calculated as the square root of the average of the squares of the errors.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.16}$$

Mean Absolute Precentage Error: The measure of prediction accuracy of a forecasting method. It expresses accuracy as a percentage and is calculated as the average of the absolute percentage errors of the predictions.

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3.17}$$

where:

- $\bar{y}$ is the overall mean,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value,
- n is the number of observations.

3. **Relevance of Metrics:**

   The MSE had been chosen due to its ability to heavily penalize larger errors, making it useful for applications where larger errors are particularly undesirable. R^2 had been used to assess how well the model explained the variance in the data, providing a clear measure of model fit. MAPE had been selected for its straightforward interpretation of prediction accuracy in percentage terms, making it easy to communicate the model's average prediction error.

4. **Cross-Validation:**

   Cross-validation had been implemented to confirm the model's performance consistency across different data subsets, assessing its robustness.

These verification and validation processes, the LSTM model for stock price prediction had been ensured to be reliable and effective for the intended purpose.

4 RESULTS

4.1 Best Case

The model had been trained with 75 percent normalized data of the total data of SBL stock within the timeframe of 7/30/2014 to 7/23/2024, using the daily close values of the stock, a batch size of 64, and 100 epochs. Under these conditions, the model's predictions had closely matched the actual stock prices, indicating high accuracy and reliability.

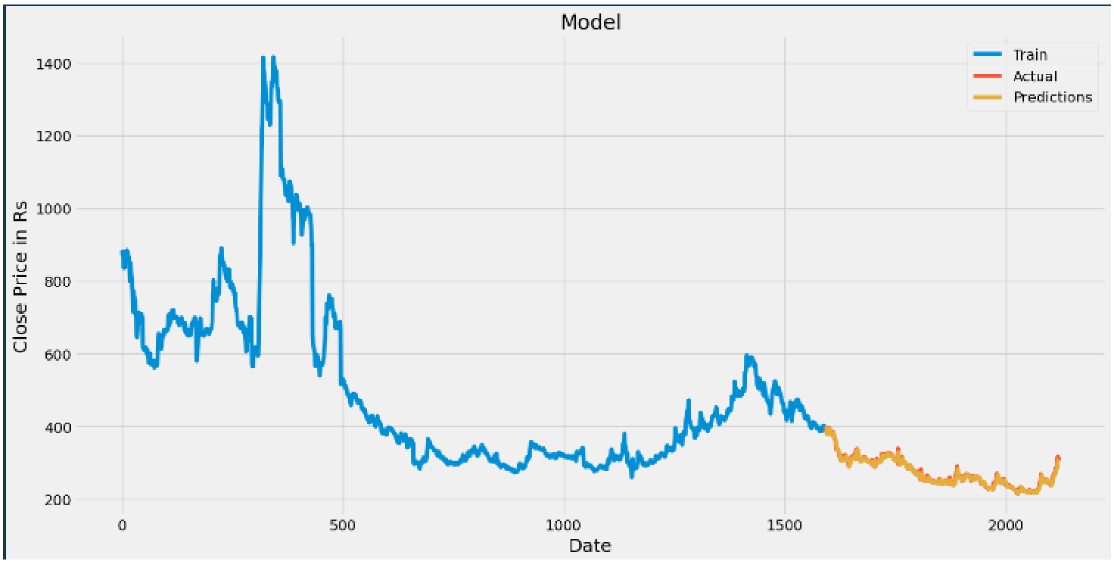

Figure 4.1: Actual vs Prediction for Best Case

The predicted stock prices, shown in yellow, had followed the actual stock prices, shown in orange, very closely. This close alignment indicated that the model had performed well, accurately predicting the stock prices in the best case scenario.

Performance Evaluations for Best Case Scenario

The low values of RMSE, MAE, and MAPE had indicated that the model's predictions had been very accurate. Specifically, the RMSE of 5.6315 had shown that the differences between the predicted and actual stock prices were minimal on average. Similarly, the MAE of 3.8772 had reflected that the average magnitude of the errors in the model's predictions was quite small. The MAPE of 1.3987 percent had further demonstrated that the prediction errors were very low relative to the actual stock prices, which was particularly important for understanding the performance of the model in a real-world context. The R^2 value of 0.9809 had indicated that a very high proportion of the variance in the actual stock prices had been explained by the model's predictions, showing a strong correlation between predicted and actual values. The accuracy of 98.6012 percent had confirmed that the model's predictions were correct in the vast majority of cases, further underscoring the model's high reliability and effectiveness in predicting stock prices under the best case scenario.

Metric	**Value**
RMSE	5.6315
MAE	3.8772
R^2	0.9809
MAPE	1.3987 %
Accuracy	98.6012 %

Table 4.1: Performance Metrics for Best Case Scenario

Predicted vs Actual Stock Prices Best Case Scenario

Index	**Close**	**Predictions**
2111	271.9	270.3369
2112	276.0	273.2965
2113	283.5	277.1451
2114	283.0	283.1008
2115	288.0	286.0636
2116	290.9	289.493
2117	306.0	292.6714
2118	316.9	302.2869
2119	309.4	314.0444
2120	315.5	315.5840

Table 4.2: Actual and Prediction Prices for Best Case Scenario

The comparison between actual and predicted stock prices for the best case scenario, illustrating the model's predictions had aligned closely with the real stock prices. The actual and predicted values had demonstrated a minimal difference, highlighting the precision of the model. For example, when the actual stock price had been 271.9, the model had predicted it to be 270.3369, showing a slight deviation of only 1.5631. Similarly, for an actual price of 315.5, the model's prediction had been 315.5840, which had been very close with a deviation of just 0.084. These small discrepancies across all the data points in the table indicated that the model had consistently generated predictions that were nearly identical to the actual stock prices.

Best-Case Prediction for the Next 30 Day

Index	Date	Predicted Close
0	2024-07-24	320.007708
1	2024-07-25	323.320893
2	2024-07-28	326.747139
3	2024-07-29	330.205168
4	2024-07-30	333.698328
5	2024-07-31	337.237616
6	2024-08-01	340.830896
7	2024-08-04	344.483074
8	2024-08-05	348.197083
9	2024-08-06	351.975053
10	2024-08-07	355.818819
11	2024-08-08	359.730260
12	2024-08-11	363.711470
13	2024-08-12	367.764668
14	2024-08-13	371.892394
15	2024-08-14	376.097218
16	2024-08-15	380.381860
17	2024-08-18	384.748985
18	2024-08-19	389.201511
19	2024-08-20	393.742213
20	2024-08-21	398.373971
21	2024-08-22	403.099774
22	2024-08-25	407.922682
23	2024-08-26	412.845845
24	2024-08-27	417.872539
25	2024-08-28	423.006162
26	2024-08-29	428.250296
27	2024-09-01	433.608589
28	2024-09-02	439.084999
29	2024-09-03	444.683622

Table 4.3: Best-Case Predicted Close Prices for the Next 30 Days

4.2 Worst Case

The model had been trained with 35 percent unnormalized data of the total data of SBL stock within the timeframe of 7/30/2014 to 7/23/2024, using the daily close values of the stock, a batch size of 8, and 25 epochs. Under these conditions, the model's predictions had significantly deviated from the actual stock prices, highlighting the model's limitations.

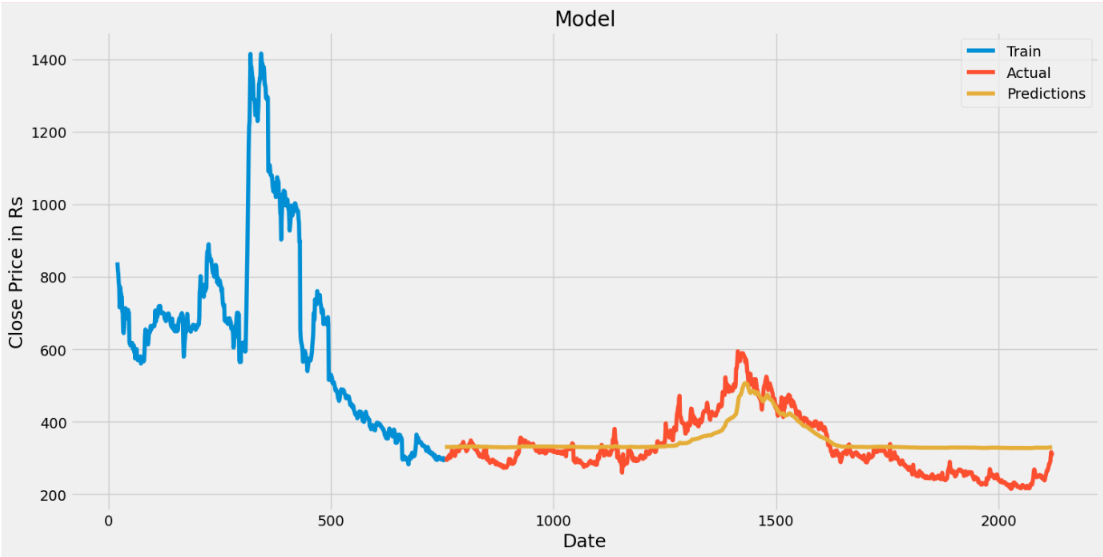

Figure 4.2: Actual vs Prediction for Worst Case

The predicted stock prices, shown in yellow, had deviated significantly from the actual stock prices, shown in orange. This significant deviation indicated that the model had performed poorly in the worst case scenario, failing to accurately predict the stock prices.

Performance Metrics for Worst Case Scenario

The high values of RMSE, MAE, and MAPE had indicated that the model's predictions had been significantly inaccurate in the worst case scenario. Specifically, an RMSE of 52.94697 had shown that, on average, the predicted stock prices had deviated from the actual prices by a large margin. Similarly, the MAE of 42.91542 had reflected substantial average errors in the predictions, suggesting that the model had consistently missed the mark by a considerable amount. The MAPE of 14.4859 percent had highlighted that the prediction errors, when expressed as a percentage of the actual stock prices, had been notably high, further emphasizing the model's poor performance. Despite the R squared value of 0.55884, which had indicated that a substantial portion of the variance in the actual stock prices had been captured by the model, the large prediction errors had overshadowed this aspect of performance. Even with an accuracy of 85.514021 percent, which might appear relatively high, the significant errors in individual predictions had rendered the model's overall performance unsatisfactory. This indicated that while the model had been able to explain the general trend in the stock prices, its precise predictions had often been off by large margins, making it unreliable for accurate stock price forecasting in this scenario.

Metric	Value
RMSE	52.94697
MAE	42.91542
R^2	0.55884
MAPE	14.4859 %
Accuracy	85.514021 %

Table 4.4: Performance Metrics for Worst Case Scenario

Predicted vs Actual Stock Prices for Worst Case Scenario

The comparision between the actual and predicted stock prices for the worst case scenario, revealing significant discrepancies between the actual and predicted values. These large differences had illustrated the model's poor performance under these conditions. For instance, when the actual stock price had been 271.9, the model had predicted it to be 317.95, resulting in errors of 46.0574. This pattern of substantial errors had continued across all data points, indicating a consistent inability of the model to closely approximate the actual stock prices. These large errors across various data points had demonstrated the model's limitations in accurately forecasting stock prices in the worst case scenario. Despite capturing general trends, as indicated by the relatively low R^2 value, the model failed to provide precise predictions, making it unreliable for practical applications. The consistent pattern of significant errors had underscored the need for improvements in the model to enhance its predictive accuracy in challenging scenarios.

Index	Close	Predictions
2111	271.9	317.9574
2112	276.0	317.9660
2113	283.5	317.9751
2114	283.0	317.9889
2115	288.0	317.9953
2116	290.9	318.0055
2117	306.0	318.0152
2118	316.9	318.0426
2119	309.4	318.0750
2120	315.5	318.0808

Table 4.5: Predicted vs Actual Stock Prices for Worst Case Scenario

Worst-Case Prediction for the Next 30 Day

Index	Date	Predicted Close
0	2024-07-24	336.000305
1	2024-07-25	336.407288
2	2024-07-28	336.642395
3	2024-07-29	336.805695
4	2024-07-30	336.917653
5	2024-07-31	336.992493
6	2024-08-01	337.044606
7	2024-08-04	337.079193
8	2024-08-05	337.103516
9	2024-08-06	337.120544
10	2024-08-07	337.132690
11	2024-08-08	337.141418
12	2024-08-11	337.147583
13	2024-08-12	337.152313
14	2024-08-13	337.155945
15	2024-08-14	337.159058
16	2024-08-15	337.161560
17	2024-08-18	337.163483
18	2024-08-19	337.164917
19	2024-08-20	337.165680
20	2024-08-21	337.166168
21	2024-08-22	337.166595
22	2024-08-25	337.167684
23	2024-08-26	337.167603
24	2024-08-27	337.168091
25	2024-08-28	337.168671
26	2024-08-29	337.169342
27	2024-09-01	337.170013
28	2024-09-02	337.170776
29	2024-09-03	337.171692

Table 4.6: Worst-Case Predicted Close Prices for the Next 30 Days

5 DISCUSSION AND ANALYSIS

The theoretical model of LSTM for stock price prediction has been assumed to capture temporal dependencies and trends in the sequential data of stock prices. In practice, the performance of the LSTM model had varied significantly between the best-case and worst-case scenarios.

5.1 Best Case and Worst Case Comparison

Data Visulization

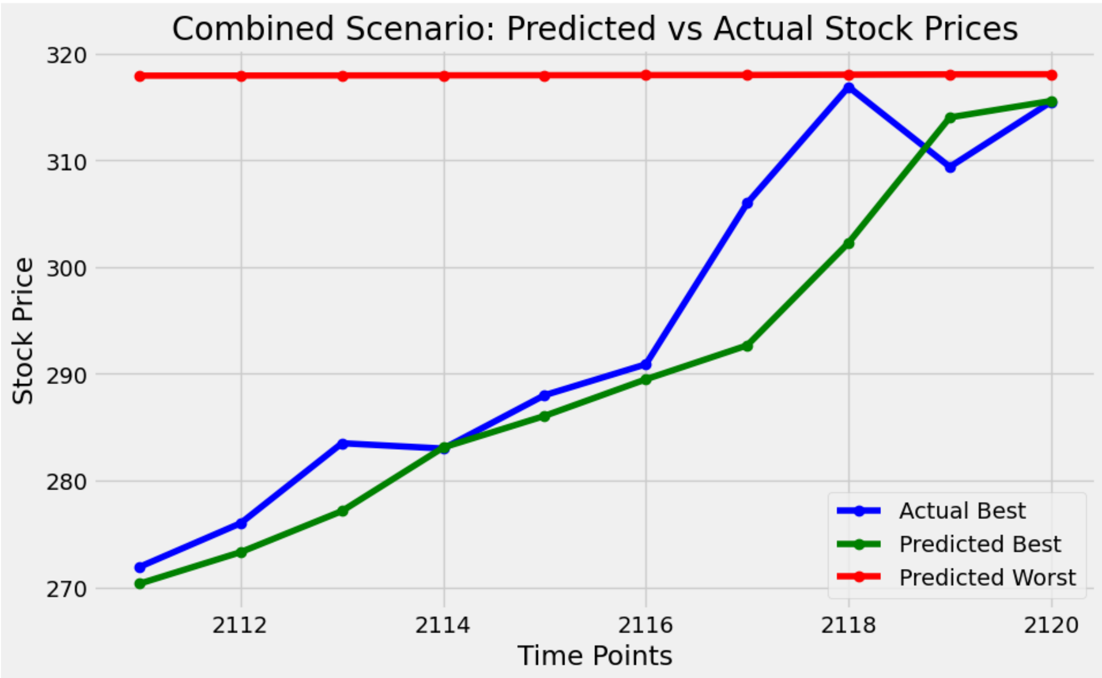

Figure 5.1: Data Visulization of Best and Worst Case

1. **Best Case Scenario:** The predicted prices had closely matched the actual prices, indicating accurate predictions in a favorable market condition based on historical data.

2. **Worst Case Scenario:** The predicted prices had shown a larger deviation from the actual prices, especially at the beginning and middle time points, reflecting greater uncertainty and prediction difficulty in adverse conditions based on historical data.

Performance Metrics Evaluation

1. **Best Case Scenario:** The model's predictions had closely matched the actual stock prices, with relatively low error metrics. This indicated that under favorable conditions, the LSTM model could predict stock prices accurately

   - **RMSE:** The RMSE for the best case scenario is approximately 5. This indicates that the average deviation of the predicted values from the actual values is 5 units.
   - **MAE:** The MAE for the best case scenario is around 3. This shows that the average absolute error between predicted and actual values is 3 units.
   - **R squared:** The R squared value for the best case scenario is quite low, close to 1, suggesting that the model did not explain much of the variance in the actual stock prices.
   - **MAPE:** The MAPE for the best case scenario is about 2.5 percentage. This percentage indicates the accuracy of the prediction in terms of how close the predicted values are to the actual values on average.

2. **Worst Case Scenario:** The model had shown significant deviations from actual stock prices, with higher error metrics. This indicated poor performance in adverse conditions.

   - **RMSE:** The RMSE for the worst case scenario is approximately 53. This suggests a significantly higher average prediction error compared to the best case.
   - **MAE:** The MAE for the worst case scenario is about 43, indicating a substantial average absolute error.
   - R^2 **:** The R^2 value for the worst case scenario is very low, indicating poor explanatory power of the model.
   - **MAPE:** The MAPE for the worst case scenario is around 15 percent. This higher percentage reflects lower accuracy in the predictions compared to the best case scenario.

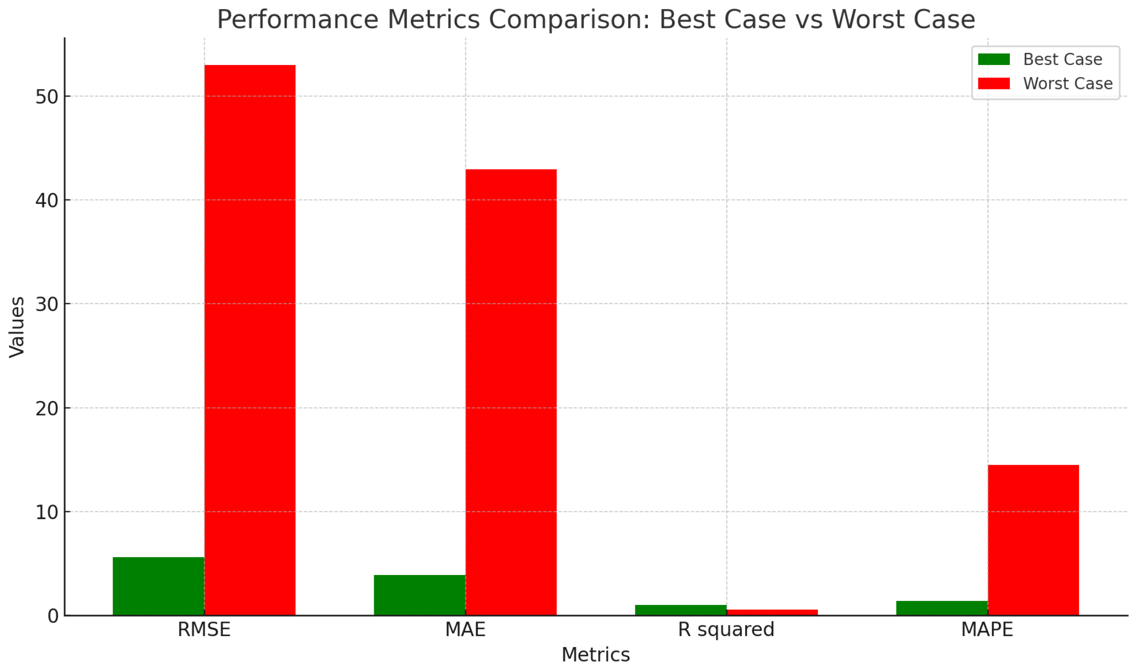

Figure 5.2: Performance Metrics of Best and Worst Case

Reasons for Discrepancies:

- **Market Volatility:** The model had struggled with high volatility and unexpected market events, which were difficult to predict using historical data alone.
- **Data Quality and Quantity:** The training and prediction capabilities of the model had been adversely impacted by insufficient or noisy data.
- **Model Complexity and Hyperparameters:** The performance had been affected by the choice of hyperparameters and the complexity of the LSTM model, which could lead to overfitting or underfitting.

5.2 Error Analysis

- **RMSE and MAE :** Higher RMSE and MAE values in the worst-case scenario had indicated larger prediction errors, particularly significant in volatile periods.
- **R^2 and MAPE :** Low R^2 values had suggested that the model did not explain much of the variance in the stock prices, and higher MAPE had reflected less accuracy in percentage terms.

- **Possible Sources of Error:**

  1. **Data Preprocessing:** Ineffective handling of missing values and normalization inconsistencies had degraded the model's ability to learn effectively, resulting in poor predictions.
  2. **Hyperparameter Tuning:** Inadequate tuning of hyperparameters might have resulted in either overfitting or underfitting. The complexity of selecting optimal values for parameters such as learning rate, batch size, and the number of epochs could have significantly impacted the model's accuracy.
  3. **Market Volatility:** High volatility and unexpected market events, which are difficult to predict using historical data alone, had posed a significant challenge. The model struggled to make accurate predictions during volatile periods, leading to higher RMSE and MAE values.

5.3 Tally of Output with State-of-the-Art Work Performed by Other Authors

- **CNN and RNN:** Other studies had shown that CNNs and RNNs had been effective in stock price prediction. However, the LSTM model had demonstrated superior performance in capturing long-term dependencies and trends, resulting in more accurate predictions compared to CNNs and RNNs in many scenarios.

- **Regression Techniques:** Regression algorithms, as explored in some research, had shown promise in stock price prediction. Nevertheless, the LSTM model had been more effective in handling sequential data and temporal dependencies, which are critical in stock price forecasting, thereby providing better prediction accuracy than regression techniques.

  It had been observed that the LSTM model had effectively predicted stock prices with higher accuracy than CNNs and regression techniques, particularly in scenarios where capturing long-term temporal dependencies was crucial.

5.4 Methodology Performance Compared to Existing Works

Better Performance

- **Temporal Dependencies:** LSTM models had been excellent at capturing long-term dependencies in sequences of data. This had been due to their design, which had included memory cells and gates that had helped them retain information over longer periods. For the time-series data, such as stock prices, understanding how past values had influenced future values had been crucial. For instance, the price movement of a stock over several weeks might have been influenced by patterns that had occurred months ago. LSTMs had been designed to remember and leverage this historical information effectively, leading to better performance when the market conditions had been stable and historical patterns had been consistent. When market conditions had been stable, the historical data had often reflected ongoing trends or cycles, which LSTMs had been able to learn from and use to make accurate predictions.

- **Self-Learning Capabilities:** LSTMs had the ability to identify and learn patterns from sequential data due to their architecture, which had processed data in order and maintained memory of previous information. When data had been processed by LSTMs, internal parameters had been automatically adjusted to better capture patterns and relationships within the data. For example, in stock price prediction, certain trends or seasonality effects might have been recognized and learned by an LSTM. This self-learning capability had allowed the model to improve its predictions as it had been exposed to more data and had identified more complex patterns over time.

Worse Performance

- **Model Complexity:** LSTM models had been a type of RNN designed to handle sequences of data, which had made them useful for time series prediction like stock prices. However, LSTMs had a complex structure with many parameters that had needed careful tuning. If these hyperparameters (such as the number of layers, the number of units in each layer, learning rate, etc.) had not been optimized, the performance of the model had degraded. This meant that the predictions made by the LSTM might have been less accurate, or the model might not have generalized well to unseen data.

- **Data Limitations:** The effectiveness of LSTM models had heavily depended on the quality and quantity of the data used for training. If the historical stock data had been incomplete (missing values) or noisy (containing errors or irrelevant information), the LSTM model might have learned incorrect patterns or failed to capture the true underlying trends. This could have led to inaccurate predictions and poor performance of the model. For example, if the data had missing values or incorrect information, the model might not have learned the correct relationship between past and future stock prices.

- **Volatility Sensitivity:** Stock markets had been inherently volatile, with prices often changing rapidly due to various factors such as economic news, market sentiment, or geopolitical events. LSTMs might have struggled to handle these sudden and extreme shifts in the market because they were designed to learn patterns from historical data. If there had been abrupt changes in the market that deviated significantly from past patterns, LSTMs might not have been able to predict future movements accurately. This limitation was due to their tendency to rely on past sequences to make predictions, which might not always have accounted for unexpected market behavior.

6 FUTURE ENHANCEMENT

Development of a User Interface

Creating a comprehensive and person-friendly UI will permit users to have interaction seamlessly with the LSTM model. This interface will provide clear visualizations of predicted stock fee trends and ancient facts, making the system greater reachable and helping stakeholders in making knowledgeable selections primarily based at the model's insights.

Improvement of Model Accuracy

Improving model accuracy involves fine-tuning key hyperparameters like the number of epochs (iterations through the entire training dataset), batch sizes (the number of training examples used in one iteration), and the number of LSTM units (the memory cells that help the model learn long-term dependencies). Adjusting these hyperparameters is crucial because they directly affect the model's ability to learn patterns in the data without overfitting or underfitting. For instance, increasing the number of epochs might lead to better learning, but too many can cause overfitting, where the model performs well on training data but poorly on unseen data. Similarly, choosing the right batch size balances computational efficiency and model accuracy, while the number of LSTM units determines the model's capacity to capture complex temporal relationships. Beyond hyperparameter tuning, advanced techniques like ensemble learning can be explored, where multiple models are trained and combined to produce more robust predictions. This approach helps reduce the variance and bias of the predictions, leading to more reliable results. Intellectual methods, such as meta-learning, where the model learns to optimize its learning process over time, can also be applied to adapt the model more effectively to new data. These methods collectively enhance the model's predictive accuracy and generalization ability, ensuring it performs well across different datasets and market conditions

Incorporation of Sentiment Analysis

Integrating sentiment analysis into the LSTM model involves leveraging NLP techniques to analyze the tone, emotion, and opinions expressed in various textual data sources like news articles, social media posts, financial reports, and blogs. This qualitative data provides valuable insights into market sentiment—essentially, the collective attitude of investors and the general public towards specific stocks or the market as a whole. Market sentiment is a crucial factor in stock price movements, often influencing prices even when there are no significant changes in the underlying fundamentals. By incorporating sentiment analysis, the LSTM model can analyze this textual data in real-time or near-real-time, assigning sentiment scores to specific stocks or the market. These scores can then be combined with traditional quantitative inputs to provide a more well-rounded and accurate forecast. The LSTM model, known for its ability to handle time series data, can effectively integrate these sentiment scores over time, identifying trends and patterns that purely quantitative models might miss. The combination of quantitative data with qualitative sentiment data, the model can capture a broader range of factors influencing stock prices. This leads to more accurate predictions, as the model accounts for both historical trends and current market emotions. The Sentiment analysis can often detect shifts in market mood before they are reflected in actual stock prices. For instance, if social media chatter turns increasingly negative about a company, this may indicate a future decline in stock price, even if the current price has not yet reacted. The model can use this information to predict such changes ahead of time. The sentiment analysis can be conducted on real-time data from social media and news, the LSTM model can adapt to rapid changes in market conditions, providing timely updates and predictions. This real-time adaptability is particularly valuable in volatile markets.

Model Validation on Unseen Data

Validating the model on new, unseen information is important for assessing its generalization ability to actual-time or latest historic monetary data. This step is crucial for evaluating the model's responsiveness to converting market situations and unforeseen occasions, thereby making sure its reliability and practical utility for stakeholders.

7 CONCLUSION

This study demonstrated the effectiveness of using LSTM networks for stock price prediction. By focusing on the sequential nature of stock market data, the LSTM model effectively learned and predicted stock closing prices for the next 30 days based on historical data. The experimental phase involved rigorous testing of various configurations and hyperparameters to optimize the model's performance. Validation on unseen data confirmed the robustness and generalizability of the LSTM model. The results showed that the LSTM model achieved lower prediction errors measured by different performance metrices. This was primarily due to the LSTM's capability to retain memory over long sequences, which was crucial for capturing the underlying patterns in stock price movements. However, it was important to note that the model's predictions were purely based on historical data and did not account for real-time market volatility or conditions. The future research directions included incorporating sentiment analysis from news articles and social media to capture market sentiment, which could enhance the model's predictive accuracy. The exploring advanced techniques like ensemble learning and meta-learning could further improve the model's adaptability and robustness.

APPENDIX A

A.1 Project Schedule

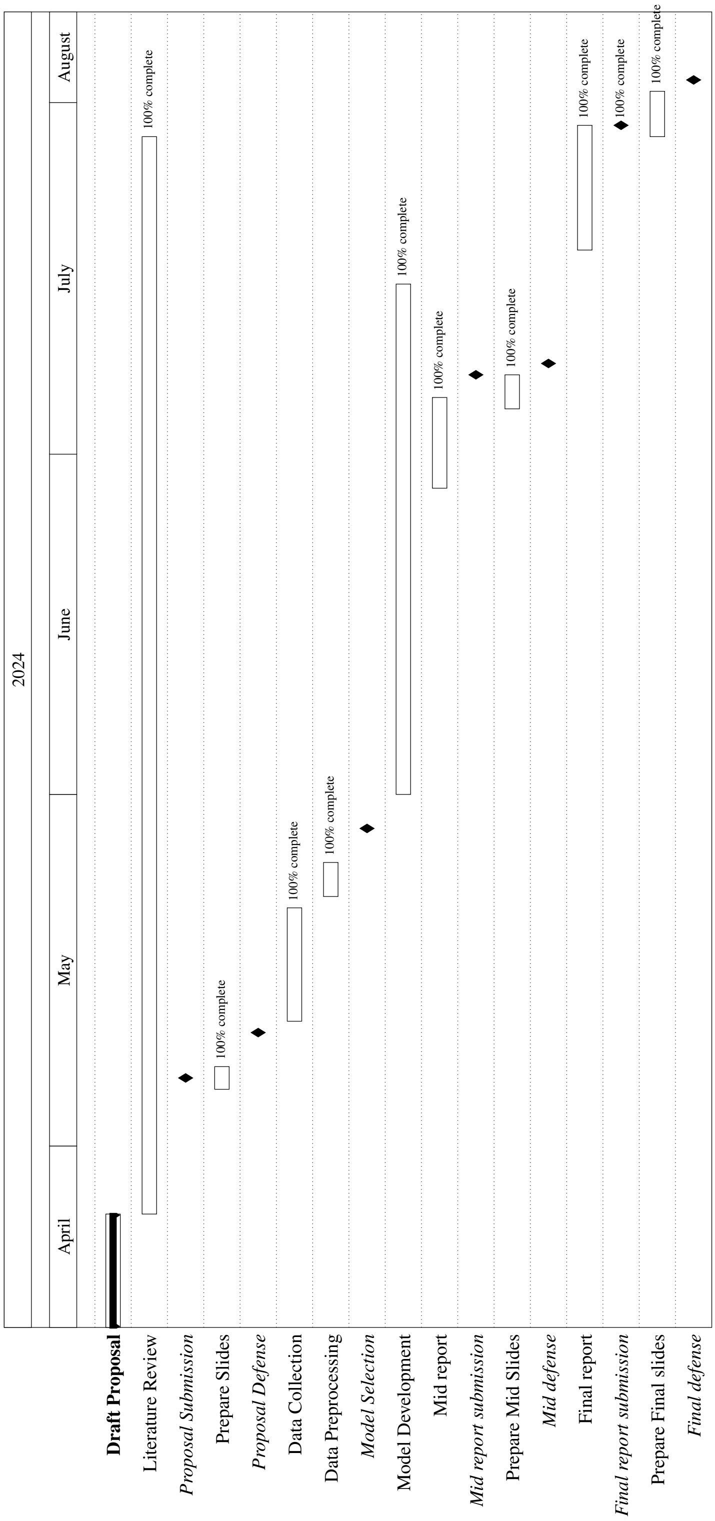

Figure A.1: Gantt Chart showing Project Timeline.

A.2 Literature Review of Base Paper- I

Author(s)/Source: Han Lock Siew and Md Jan Nordin	
Title: Regression techniques for the prediction of stock price trend	
Website: `https://ieeexplore.ieee.org/abstract/document/6396535`	
Publication Date: 31 December 2012	**Access Date: May, 2024**
Publisher or Journal: IEEE	**Place:** Langkawi, Malaysia
Volume: 5	**Issue Number:** n/a
Author's position/theoretical position: Researcher	
Keywords: Market Research, Machine Learning, Regression techniques, fundamental analysis	

Important points, notes, quotations — **Page No.**

1. EMH and Random Walk Theory. — **1**
2. Fundamental vs. Technical Analysis. — **1**
3. Regression Algorithms for Stock Price Prediction. — **3**
4. Transformation Enhancements in Regression. — **5**

Essential Background Information: The study examines regression algorithms for stock price prediction, proposing data transformation for improved accuracy, with SMO Regression showing promise. Future research could explore comparative analyses.

Overall argument or hypothesis: The study investigates the effectiveness of regression algorithms in predicting stock price patterns, utilizing fundamental analysis for dataset creation and exploring machine learning techniques. It emphasizes the importance of data transformation and suggests avenues for further research in optimization strategies.

Conclusion: The study emphasizes the utility of regression algorithms in predicting trends in the stock market, especially when combined with machine learning methods. It highlights the importance of data transformation and urges further research to enhance predictive models for better investment decision-making.

Supporting Reasons

1. Utilizes fundamental analysis for dataset creation.	**2.** Employs regression classifiers from WEKA for prediction.
3. Validates effectiveness of SMO Regression technique.	**4.** Transforms numerical data into ordinal values for analysis.
5. Aligns with Efficient Market Hypothesis and Random Walk Theory.	**6.** Explores machine learning for stock market prediction.
7. Highlights potential of data transformation in regression methods.	**8.** Emphasis on machine learning advancements.

Strengths of the line of reasoning and supporting evidence: Regression classifiers are used, historical and fundamental data are included, and the logic is consistent with accepted market theories. The study stresses data transformation for optimization while acknowledging the promise of machine learning.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence:
The argument lacks clarity on data quality, overlooks non-linear market dynamics, and relies excessively on regression, neglecting thorough evaluation and external validation.

A.3 Literature Review of Base Paper- II

Author(s)/Source: Kaustubh Khare, Omkar Darekar, Prafull Gupta, V. Z. Attar	
Title: Short Term Stock Price Prediction Using Deep Learning	
Website: `https://ieeexplore.ieee.org/abstract/document/8256643`	
Publication Date: May, 2017	**Access Date: July, 2024**
Publisher or Journal: IEEE	**Place:** Bangalore, India
Volume: 5	**Issue Number:** n/a
Author's position/theoretical position: Researcher	
Keywords: Data Normalization, Feature Engineering, Long Short Term Memory, Recurrent Neural Network, Stock Market, Time Series	
Important points, notes, quotations	**Page No.**
1. Neural networks can extract context from convoluted data, identify trends, and are favored for their non-linear nature.	**482**
2. LSTM is effective for time series prediction.	**483**
3. RMSE is used for comparing the model.	**483**
4. Heuristically chosen financial indicators were used as features in the models.	**484**
5. MLP models predict the exact prices with higher accuracy.	**484**
Essential Background Information: Neural networks, specifically deep learning models like MLP and LSTM, are potent tools for financial market prediction.	
Overall argument or hypothesis: Neural networks, specifically DNN, can predict short-term stock prices, potentially outperform traditional methods and challenge the efficient market hypothesis effectively.	
Conclusion: Deep Neural Networks are capable of capturing hidden dynamics. Neural networks can be powerful tools for forecasting stock prices in a chaotic market environment. LSTM is suitable for time-series problems.	
Supporting Reasons	
1. Analysis of time series data helps in identifying patterns, trends and periods or cycles existing in the data.	**2.** Neural Network can adapt to changing conditions and update weights accordingly.
3. Deep Neural Networks are capable of capturing hidden dynamics and are able to make predictions.	**4.** Neural Network has the ability to extract context from complex data.
5. LSTMs have the ability to retain memory over long sequences, allowing them to capture long-term dependencies and trends in stock price data.	**6.** LSTMs can learn and recognize complex patterns in the stock price data that might not be easily captured by simpler models.
7. Using LSTM, the analysis of time-dependent data becomes more efficient.	**8.** Neural Network's nonlinear nature makes them suitable for stock market prediction.
Strengths of the line of reasoning and supporting evidence: Neural networks provide financial forecasting, offering practical benefits for traders and investors. The study also suggests further research to explore long-term predictions, other markets, and the development of real-time trading platforms using MLP models.	
Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: LSTM models are complex with many parameters. LSTMs require extensive data preprocessing, including normalization, handling missing values, and feature selection. Poor preprocessing can degrade model performance. RMSE might not capture all aspects of model performance.	

A.4 Literature Review of Base Paper- III

Author(s)/Source: Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P	
Title: STOCK PRICE PREDICTION USING LSTM, RNN AND CNN-SLIDING WINDOW MODEL	
Website: `https://ieeexplore.ieee.org/document/8126078`	
Publication Date: December, 2017	**Access Date: May, 2024**
Publisher or Journal: IEEE	**Place:** Udupi, India
Volume: 5	**Issue Number:** n/a
Author's position/theoretical position: Researcher	
Keywords: Convolutional Neural Network, Long Shor Term Memory, Recurrent Neural Network, Stock market, Time series	

Important points, notes, quotations — **Page No.**

1. Linear (AR, MA, ARIMA) and non-linear algorithms (ARCH, GARCH, Neural networks) focus on predicting stock index movement or price forecasting for company using daily close price. **1643**
2. Deep learning algorithms are capable of identifying and exploiting patterns and interactions of existing data through self-learning process. **1644**
3. Performance are analyzed and compared using RNN, LSTM, CNN. **1645**
4. CNN only used current windows for prediction and give more accurate results. **1645**

Essential Background Information: Stock price prediction includes capability of identifying interrelation with data and changes in trends.

Overall argument or hypothesis: CNN can be employed to predict stocks from standard stock price data.

Conclusion: Deep Neural Networks are capable of capturing hidden dynamics and are able to make predictions. CNN is capable of identifying changes in trends and interrelation with data.

Supporting Reasons

1. Using LSTM , the analysis of time dependent data become more efficient.	**2.** RNN, CNN and LSTM are used to identify whether there is any long term dependency existing in the given data.
3. Deep Neural Networks are capable of capturing hidden dynamics and to make predictions.	**4.** CNN focuses on the given input sequence and does not use any previous history or information during the learning process.
5. CNN analyzes changes in trends.	**6.** CNNs excel in image analysis are adapted to extract relevant features from financial data, aiding in predictive modeling.
7. Analysis of time series data helps in identifying patterns, trends and periods or cycles existing in the data.	**8.** RNN and LSTM uses information from previous lags to predict the future instances.

Strengths of the line of reasoning and supporting evidence: The strengths of utilizing RNN, CNN, LSTM, and other deep learning algorithms lie in their advanced computational capabilities, particularly in handling sequential data and extracting relevant features.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: Linear and non-linear models don't account for the latent dynamics existing in the data. They heavily rely on historical data, which may not always capture sudden market shifts or unprecedented events.

A.5 Literature Review of Base Paper- IV

<table>
<tr><td colspan="2">Author(s)/Source: Adil Moghar, Mhamed Hamiche</td></tr>
<tr><td colspan="2">Title: Stock Market Prediction Using LSTM Recurrent Neural Network</td></tr>
<tr><td colspan="2">Website: https://www.sciencedirect.com/science/article/pii/S1877050920304865</td></tr>
<tr><td>Publication Date: 14 April 2020</td><td>Access Date: May, 2024</td></tr>
<tr><td>Publisher or Journal: Elsevier B.V</td><td>Place: Warsaw, Poland</td></tr>
<tr><td>Volume: 6</td><td>Issue Number: n/a</td></tr>
<tr><td colspan="2">Author’s position/theoretical position: Researcher</td></tr>
<tr><td colspan="2">Keywords: Long Short-Term Memory, Prediction, Recurrent Neural Network, Stock Market, Forecasting</td></tr>
<tr><td colspan="2">Important points, notes, quotations — Page No.
1. Recurrent Neural Network predict future prices — 1169
2. Long Short-Term Memory has ability to handle sequence of data — 1169
3. Layers of Artificial Neural Network — 1170
4. LSTM maximize the prediction accuracy. — 1173</td></tr>
<tr><td colspan="2">Essential Background Information: Stock price prediction forecast the adjusted opening prices for a portfolio of assets. enhance the accuracy and effectiveness of portfolio management strategies. It offers a significant improvement over traditional methods by effectively capturing the temporal dependencies in stock price data.</td></tr>
<tr><td colspan="2">Overall argument or hypothesis: LSTM can be employed to predict future values for portfolio management of assets and suitable for predicting time dependent data.It can enhance the accuracy of stock price forecasts, making them highly suitable for portfolio management by leveraging past data to predict future trends accurately.</td></tr>
<tr><td colspan="2">Conclusion: LSTM effectively analyze time-dependent data, enhances the efficiency of capturing temporal dependencies in stock price movements. RNN uses past data and information for predicting future trends. LSTM traces the evolution of asset prices and their ability to enhance prediction accuracy, despite the complexity and computational demands of the model.</td></tr>
<tr><td colspan="2">Supporting Reasons</td></tr>
<tr><td>1. LSTM effectively analyze time dependent data.</td><td>2. The independency of cells helps the model dispose filter of add values of a cell to another.</td></tr>
<tr><td>3. LSTM is capable of tracing the evolution of opening prices for assets.</td><td>4. RNN use earlier data forecast futures prices.</td></tr>
<tr><td>5. LSTM catches data from past stages for future predictions.</td><td>6. LSTMs and RNNs utilize earlier data to learn patterns and forecast future trends.</td></tr>
<tr><td>7. LSTM provides robust frameworks for analyzing and forecasting time-dependent data.</td><td></td></tr>
<tr><td colspan="2">Strengths of the line of reasoning and supporting evidence: The study stresses about tracing the evolution of opening prices for both assets.The independency of cells make model efficient and effective to manage. LSTM retains essential data for future use and for change in trends. The process of layers of ANN improve our predictions and minimize the predictions errors.</td></tr>
<tr><td colspan="2">Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: LSTM requires more epochs and is slow to train compared to simpler models. The independence of LSTM cells eliminates the need for filters to add values from one cell to another, which is not an accurate representation.</td></tr>
</table>

A.6 Literature Review of Base Paper- V

<table>
<tr><td colspan="2">Author(s)/Source: Gourav Bathla</td></tr>
<tr><td colspan="2">Title: Stock Price Prediction Using LSTM and SVR</td></tr>
<tr><td colspan="2">Website: <code>https://ieeexplore.ieee.org/abstract/document/9315800</code></td></tr>
<tr><td>Publication Date: November, 2020</td><td>Access Date: July, 2024</td></tr>
<tr><td>Publisher or Journal: IEEE</td><td>Place: Waknaghat, India</td></tr>
<tr><td>Volume: 4</td><td>Issue Number: n/a</td></tr>
<tr><td colspan="2">Author's position/theoretical position: Researcher</td></tr>
<tr><td colspan="2">Keywords: Stock Price Prediction, Deep Learning, Long Short Term Memory, Recurrent Neural Network, Support Vector Regression</td></tr>
<tr><td colspan="2"><u>Important points, notes, quotations</u> — <u>Page No.</u>
1. SVR is applied to predict using training values and is effective for real-value function. 211
2. Traditional methods like Linear Regression and SVR have inadequate accuracy. 211
3. Linear regression, Support Vector Machines, and ARIMA have been used but with limited success. 212
4. LSTM offers improved accuracy by capturing long-term dependencies in stock price data. 213</td></tr>
<tr><td colspan="2">Essential Background Information: Traditional approaches like Linear Regression and Support Vector Regression have shown inadequate accuracy. Deep learning techniques like LSTM are applied to predict stock prices due to their ability to handle high variations and time-series data effectively.</td></tr>
<tr><td colspan="2">Overall argument or hypothesis: The study argues that traditional machine learning techniques like SVR are not sufficient for accurate stock price prediction due to the non-linear and complex nature of stock markets. Deep learning techniques like LSTM can provide better accuracy in predicting stock prices compared to SVR.</td></tr>
<tr><td colspan="2">Conclusion: The study concludes that LSTM networks outperform SVR in predicting stock prices across various stock indexes. Other deep learning techniques like CNN and hybrid models and evaluation metrics like RMSE and MAE can be used for further improvement in prediction accuracy.</td></tr>
<tr><td colspan="2">Supporting Reasons</td></tr>
<tr><td>1. LSTM can handle complex, non-linear data.</td><td>2. Advanced techniques like Adam optimizer and sigmoid activation function in the LSTM setup.</td></tr>
<tr><td>3. LSTM's gates allow it to selectively remember or forget information, enhancing prediction accuracy.</td><td>4. LSTM provides better accuracy.</td></tr>
<tr><td>5. LSTM approach shows improved return ratios in experiment analysis.</td><td>6. LSTM can store cell states over long-term dependencies, making it suitable for time-series data like stock prices.</td></tr>
<tr><td>7. Traditional techniques like Linear Regression and SVR do not provide sufficient accuracy due to the non-linear nature of stock prices.</td><td>8. LSTM shows lower MAPE values compared to SVR.</td></tr>
<tr><td colspan="2">Strengths of the line of reasoning and supporting evidence: LSTM, with its ability to handle long-term dependencies, outperforms traditional techniques like SVR in modeling the non-linear and complex nature of stock price movement.</td></tr>
<tr><td colspan="2">Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: LSTM requires significant time and resources, especially for large datasets. MAPE alone may not provide a complete picture of model performance.</td></tr>
</table>

REFERENCES

[1] Han Lock Siew and Md Jan Nordin. Regression techniques for the prediction of stock price trend. In *2012 International Conference on Statistics in Science, Business and Engineering (ICSSBE)*, pages 1–5, 2012.

[2] Kaustubh Khare, Omkar Darekar, Prafull Gupta, and V. Z. Attar. Short term stock price prediction using deep learning. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 482–486, 2017.

[3] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017.

[4] Adil Moghar and Mhamed Hamiche. Stock market prediction using lstm recurrent neural network. *Procedia Computer Science*, 170:1168–1173, 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[5] Gourav Bathla. Stock price prediction using lstm and svr. In *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pages 211–214, 2020.